

UiO : Department of Mathematics  
University of Oslo

# Selection of Claim Size Distribution in General Insurance

**Simen Holter**

Master's Thesis, Spring 2017



This master's thesis is submitted under the master's programme *Modelling and Data Analysis*, with programme option *Finance, Insurance and Risk*, at the Department of Mathematics, University of Oslo. The scope of the thesis is 30 credits.

The front page depicts a section of the root system of the exceptional Lie group  $E_8$ , projected into the plane. Lie groups were invented by the Norwegian mathematician Sophus Lie (1842–1899) to express symmetries in differential equations and today they play a central role in various parts of mathematics.

# Abstract

Computations in general insurance are often based on models such as the *collective risk model*, which uses a compound distribution to describe the aggregated losses. A critical part of this model is the uncertainty of claim sizes. The claim sizes are typically modeled through simple two-parameter distributions where their fit are assessed by Q-Q plots. Another approach is to use more flexible distributions which can be fitted to different samples, everything between light-tailed and heavy-tailed distributions.

We will use an extended Pareto model with three parameters and a 4-parameter model with some of the standard two-parameter families as special cases. We use Monte Carlo-simulations to analyze how well the 3- and 4-parameter models estimate the reserve compared to the special cases Gamma, Weibull and Pareto distribution. More parameters provide a more flexible model, but it also means that the uncertainty becomes larger in the reserve estimate. We use error analysis to determine how well the models performs for the different distributions and for varying sample sizes. Finally, we find that the 3- and 4-parameter models provide a good fit for sample sizes  $n = 5000$  and  $n = 500$ , and partly for  $n = 50$ . We find that the 4-parameter model is superior to the 3-parameter model. Also, the 4-parameter model is slightly overestimating the reserve which makes the 4-parameter model a safe and conservative choice for the claim size distribution.



# Acknowledgements

I would like to thank my supervisor, Ingrid Hobæk Haff, for providing me with an interesting and exciting topic. I appreciate your feedback, contributions and valuable comments during the writing process. In addition, thank you for the method of moments R-code.

I would also like to thank my family for their support and encouragement, and for the time you have taken to proofread my thesis. Finally, I would like to thank my significant other for keeping me motivated throughout this period.

*Simen Holter*  
May 2017, Oslo

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Notation and Theory</b>	<b>3</b>
2.1	Parameter Estimation . . . . .	3
2.1.1	The Method of Moments . . . . .	4
2.1.2	Maximum Likelihood . . . . .	4
2.2	Collective Risk Model . . . . .	5
2.3	Claim Frequency Modeling . . . . .	6
2.3.1	The Poisson Distribution . . . . .	6
2.4	Claim Size Modeling . . . . .	8
2.4.1	Parametric Distributions . . . . .	8
2.4.1.1	The Gamma Distribution . . . . .	9
2.4.1.2	The Weibull Distribution . . . . .	11
2.4.1.3	The Pareto Distribution . . . . .	12
2.4.1.4	The Extended Pareto Distribution . . . . .	15
2.4.1.5	4-parameter Distribution . . . . .	17
2.4.2	Non-parametric Model . . . . .	19
2.5	Reserve . . . . .	20
2.5.1	Monte Carlo Simulation . . . . .	20
2.5.2	Estimation of the Reserve . . . . .	21
<b>3</b>	<b>Simulation</b>	<b>23</b>
3.1	Plan for the Simulation Study . . . . .	23
3.2	The model . . . . .	24
3.3	Parameter Setting . . . . .	25
<b>4</b>	<b>Reserve</b>	<b>27</b>
4.1	Error Analysis . . . . .	27
4.1.1	Sample Size: 5 000 . . . . .	28
4.1.2	Sample Size: 500 . . . . .	31
4.1.3	Sample Size: 50 . . . . .	34

4.2 Summary . . . . .	37
<b>5 Conclusion</b>	<b>39</b>
<b>Bibliography</b>	<b>41</b>
<b>Appendices</b>	<b>43</b>
<b>A Mathematical Elaborations</b>	<b>45</b>
A.1 Probability . . . . .	45
A.2 Inversion . . . . .	46
A.3 Extended Pareto: A Representation . . . . .	47
<b>B R codes</b>	<b>49</b>
B.1 R code - Chapter 2 . . . . .	49
B.2 R code - Chapter 3 . . . . .	51
B.3 R code - Chapter 4 . . . . .	60





# Chapter 1

## Introduction

Ever since Filip Lundberg introduced collective risk theory in 1909, the *collective risk model* has become the dominant model used in actuarial risk theory [Ramsay, 2008, p. 2]. It is a model for the aggregated losses, i.e. the total amount paid on all claims occurring in a fixed time period on a defined set of insurance contracts. The aggregated loss has a compound distribution with two main components. One characterizing the claim frequency and another describing the claim sizes. The claim sizes are taken to be independent, identically distributed random variables. They are also independent of the number of claims produced [Kaas et al., 2008, p. viii]. Hence, the total claims is the sum of a random number of iid individual claim amounts.

Claim sizes in non-life insurance are typically modeled through well known two-parameter distributions, such as Gamma, Weibull or Pareto, that have the best possible fit of the claim data. The goodness of fit of a distribution can be determined by statistical tests and probability plots [Arik and Umbleja, 2010, p. 28]. An alternative to this is to use richer classes of distributions with the well known models as special cases. An example is the extended Pareto model with three parameters, where the choice of parameter values can distinguish between the heavy-tailed, ordinary Pareto, the light-tailed Gamma, or something in between. In order to make a distribution more flexible, we can increase the number of parameters in the distribution. On the other hand, the uncertainty will also become larger when we estimate the aggregated loss, and furthermore the *reserve*. Regulators demand sufficient funds to cover losses with a high probability. The reserve, also called solvency capital, is an upper percentile  $q_\epsilon$  of the aggregated losses. The simplicity of a model is typically measured in terms of the number of parameters, and the conformity to data is measured in terms of the discrepancy between the data and the model. Hence, a prop-

erly constructed loss model should reflect a balance between simplicity and conformity to the data [Panjer et al., 2008, p. 3]. To examine this closer we will assess the fit of 3- and 4-parameter distributions and study their performance relative to the two-parameter special cases. We will use error analysis to evaluate how well they estimate the reserve compared to the true distributions. In addition we try different sample sizes to evaluate if these distributions are more or less desirable when data are limited.

In chapter 2, we introduce some theory and models. We will present all of the distributions used in this thesis along with their mean, standard deviation and maximum likelihood. Thereafter, the reserve will be introduced followed by an explanation of the Monte Carlo-simulation method.

In chapter 3, the model for the total loss will be constructed by utilizing theory from chapter 2. We will compute the reserves by the use of a simulation study, which includes generating random variables, estimate parameters and calculate the reserve. At the end of the chapter, a summary of the parameter setting will be given.

In chapter 4, we will perform an error analysis of the computed reserves from the simulation study. The bias and root mean square error measures will be presented and used to describe and interpret how well the 3- and 4-parameter models fits to the different distributed samples.

In chapter 5, a conclusion will be given with a discussion on how the 3- and 4-parameter models performed and what this implies.

# Chapter 2

## Notation and Theory

In the collective risk model, the aggregated losses depend on two random variables. The first describes the number of claims that will occur in a specified period. This is referred to as the frequency of claim and its probability distribution is called the frequency distribution. The second describes the amount (or size) of the claims, given that a claim has occurred. This is referred to as the severity and the probability distribution for the amount of claims is called the severity distribution. We model these quantities as random variables with appropriate probability distributions, and by combining these two distributions we can determine the overall loss distribution.

In this chapter, we will first present some methods for estimating the parameters. Then, we present some of the most common probability distributions. Finally, the reserve will be elaborated followed by an explanation of the Monte Carlo simulation method.

### 2.1 Parameter Estimation

To fit a model, we need to know how the unknown parameters of the probability distribution are estimated. This can for instance be done by the *method of moments* or the *maximum likelihood method*. The method of moments is an ancient and simple procedure, but may not always find the *best estimator*, i.e. the minimum variance unbiased estimator. Another weakness of the method of moments is that there must be as many equations as there are unknown parameters, and solutions are not necessarily unique, they may be hard to find or may not exist at all. By the maximum likelihood method, elementary differentiating can sometimes be used to derive explicit estimates of the parameters, otherwise the `optimize()` or `optim()` functions in R can be used. Theoretical properties hold the max-

imum likelihood estimates to be about the best possible [Bølviken, 2014, p. 237]. Therefore we will use the maximum likelihood method in this thesis. However, we will use the method of moments to provide starting values for the `optim()` functions.

### 2.1.1 The Method of Moments

The *method of moments* generates parameter estimates by simply matching the moments of the model,  $E(Z)$ ,  $E(Z^2)$ ,  $E(Z^3)$ , ..., in turn to the required number of corresponding sample moments calculated from the data  $z_1, \dots, z_n$ , where  $n$  is the number of observations. The sample moments are

$$\frac{1}{n} \sum_{i=1}^n z_i, \quad \frac{1}{n} \sum_{i=1}^n z_i^2, \quad \frac{1}{n} \sum_{i=1}^n z_i^3, \quad \dots \quad (2.1)$$

The equations are then solved for the parameters of interest, using the sample moments in place of the unknown moments. This will result in estimates of the parameters.

The estimates  $\tilde{\theta}$  of  $\theta$  produced are called MMEs. The method of moment approach is usually easy to implement, but may have high standard errors and in some cases the MME can be very poor and unreliable [Gray and Pitts, 2012, p. 58]. In this thesis we will use the method to provide start values for numerical optimizers that may require them.

### 2.1.2 Maximum Likelihood

The method of maximum likelihood is the most popular technique for deriving estimators. Let  $Z_1, \dots, Z_n$  have joint pdf  $f(z_1, \dots, z_n | \theta_1, \dots, \theta_m)$  where the parameters  $\theta_1, \dots, \theta_m$  have unknown values. When  $z_1, \dots, z_n$  are some observed historical data,  $f(z_1, \dots, z_n | \theta_1, \dots, \theta_m)$  is called the *likelihood function*. The idea of maximum likelihood is to adjust  $\theta_1, \dots, \theta_m$ , while  $z_1, \dots, z_n$  are kept fixed, to maximize the joint probability density function (pdf). For an independent and identically distributed sample, the likelihood function is defined by

$$L(\theta_1, \dots, \theta_m; z_1, \dots, z_n) = f(z_1, \dots, z_n | \theta_1, \dots, \theta_m) = \prod_{i=1}^n f(z_i | \theta_1, \dots, \theta_m). \quad (2.2)$$

We find the maximum likelihood estimate (MLE) by differentiating the above,  $L(\theta_1, \dots, \theta_m; z_1, \dots, z_n)$ , and setting the result equal to zero. It is

often much easier to differentiate the logarithm of the likelihood function. Therefore, we find the maximum likelihood estimate,  $\hat{\theta}_i$ , by,

$$\frac{\partial}{\partial \theta_i} \log L(\theta_1, \dots, \theta_m; z_1, \dots, z_n) |_{\theta_i = \hat{\theta}_i} = \sum_{k=1}^n \frac{\partial}{\partial \theta_i} \log f(z_k | \theta_1, \dots, \theta_m) = 0, \quad (2.3)$$

where  $i = 1, \dots, m$ . If there is no analytical expression for the maximum likelihood estimate the optimization with respect to the parameters must be done numerically, for instance using statistical software such as R.

Maximizing the likelihood gives us parameter values for which the observed sample is most likely to have been generated. When the sample size is large, the maximum likelihood estimator  $\hat{\theta}$  is close to  $\theta$ . The estimator is *consistent* which means that the estimate converges to the true value as the sample size increases. Furthermore, the MLE is often approximately unbiased,  $E[\hat{\theta}] \approx \theta$ , and has variance that is nearly as small as can be achieved by any unbiased estimator. That is, the MLE  $\hat{\theta}$  is approximately the minimum variance unbiased estimator (MVUE) of  $\theta$  [Devore and Berk, 2011, p. 357].

## 2.2 Collective Risk Model

The collective risk model is a model for the aggregated loss of an insurance portfolio. We regard a portfolio as a collective that produces a claim at random points in time. Assume there are  $J$  policies with losses  $X_1, \dots, X_J$ , and denote the claim severities by  $Z_{ji}$ . Then,

$$\mathcal{X} = X_1 + X_2 + \dots + X_J \quad \text{where} \quad X_j = Z_{j1} + Z_{j2} + \dots + Z_{jN_j}, \quad (2.4)$$

where claim numbers  $N_j$  and the losses  $Z_{ji}$  may have models depending on  $j$ . In addition, one assumes that the  $Z_{ji}$ 's are stochastically independent of  $\mathcal{N}$ , i.e. that the number of claims do not influence the severity of the claims. Although this is not always true, it is crucial for the simplicity of the model. Since we add up all losses  $Z_{ij}$ , the large and small losses will eventually even out, and we can assume that they all have a common distribution. Then the aggregated loss is

$$\mathcal{X} = Z_1 + Z_2 + \dots + Z_{\mathcal{N}}. \quad (2.5)$$

where  $\mathcal{N} = N_1 + \dots + N_J$  is the claim number at a portfolio level. The classic collective risk model turns out to be both computationally efficient and rather close to reality [Kaas et al., 2008, p. 357].

## 2.3 Claim Frequency Modeling

A critical part of risk evaluation in general insurance is the uncertainty of the claim numbers. These are well described by *counting distributions*, that is distributions of discrete random variables that can assume some or all values in  $\mathbb{N}$ . Most models for claim frequency are related to the Poisson distribution in some way. The Poisson distribution, having only one parameter to be estimated, is therefore a good choice. The parameters are  $\lambda = \mu T$  (for single policies) and  $\lambda = J\mu T$  (for portfolios) where  $J$  is the number of policies,  $\mu$  the claim intensity and  $T$  the time of exposure. If there is an over-dispersion,  $\text{Var}[N]/E[N] > 1$ , we may use the negative binomial distribution instead. In this thesis our main interest is how to model the claim sizes. Therefore, we will stick to the Poisson model when modeling claim frequency.

### 2.3.1 The Poisson Distribution

The Poisson distribution is a popular choice when modeling the counts of events that occur randomly in a given interval of time  $T$ . In order to apply the Poisson distribution, the events must be independent and occur with a known intensity.

The probability mass function (pmf) is

$$\Pr(N = n) = \frac{(\mu T)^n}{n!} e^{-\mu T}, \quad (2.6)$$

The mean and standard deviation are [Hogg and Tanis, 2010, p. 100]

$$E(N) = \lambda \quad \text{and} \quad \text{sd}(N) = \sqrt{\lambda}. \quad (2.7)$$

We assume that the number of occurrences from each policy  $j$ ,  $N_j \sim \text{Poisson}(\lambda)$  with parameter  $\lambda = \mu T$  and  $T = 1$  year. Then the *convolution property* [Bølviken, 2014, p. 283] tells us if  $N_1, \dots, N_J$  are independent and Poisson distributed with parameters  $\lambda_1, \dots, \lambda_J$ , then

$$\mathcal{N} = N_1 + N_2 + \dots + N_J \quad \sim \quad \text{Poisson}(\lambda_1 + \dots + \lambda_J), \quad (2.8)$$

We need to look at the historical data to determine the claim intensities. We let  $n_1, \dots, n_m$  be claim numbers from  $m$  policies exposed to risk during time  $T_1, \dots, T_m$ .  $\mu$  can be estimated for all  $N_j$ 's simultaneously if we assume that  $A = T_1 + \dots + T_m$  is the total risk exposure. Then we use the maximum likelihood estimation to find the estimate for  $\mu$ ,

$$L(\mu) = \prod_{i=1}^m \Pr(N = n_i | \mu) = \frac{(\mu A)^{\sum_{i=1}^m n_i} e^{-\mu A}}{\prod_{i=1}^m n_i!}. \quad (2.9)$$

The log-likelihood function is

$$\begin{aligned} \log L(\mu) &= \sum_{i=1}^m n_i \log(\mu A) - \sum_{i=1}^m \log(n_i!) - \mu A \\ &= \sum_{i=1}^m n_i \log(\mu) + \sum_{i=1}^m n_i \log(A) - \sum_{i=1}^m \log(n_i!) - \mu A. \end{aligned} \quad (2.10)$$

Differentiating with respect to  $\mu$ , we obtain the maximum likelihood estimate,

$$\frac{\partial \log L(\mu)}{\partial \mu} = \frac{\sum_{i=1}^m n_i}{\mu} - A = 0,$$

which gives

$$\hat{\mu} = \frac{n_1 + \cdots + n_m}{A}. \quad (2.11)$$

The mean and standard deviation of  $\hat{\mu}$  is [Bølviken, 2014, p. 284]

$$E(\hat{\mu}) = \mu \quad \text{and} \quad \text{sd}(\hat{\mu}) = \sqrt{\frac{\mu}{A}}. \quad (2.12)$$

We can simulate a random sample of size  $m$  from  $N \sim \text{Poisson}(\lambda)$  in R using the command `N = rpois(m, JμT)`, which will create a vector of observations of length  $m$  called  $N$ .

In Figure 2.1 the pmf of the Poisson distribution is plotted for different values of  $\lambda$ . When  $\lambda$  is small the probability mass function indicates a small number of occurrences with little variation. As  $\lambda$  becomes larger the curve is more spread out. That is, as claim intensity increases, the number of occurrences increases and they get more uncertain.

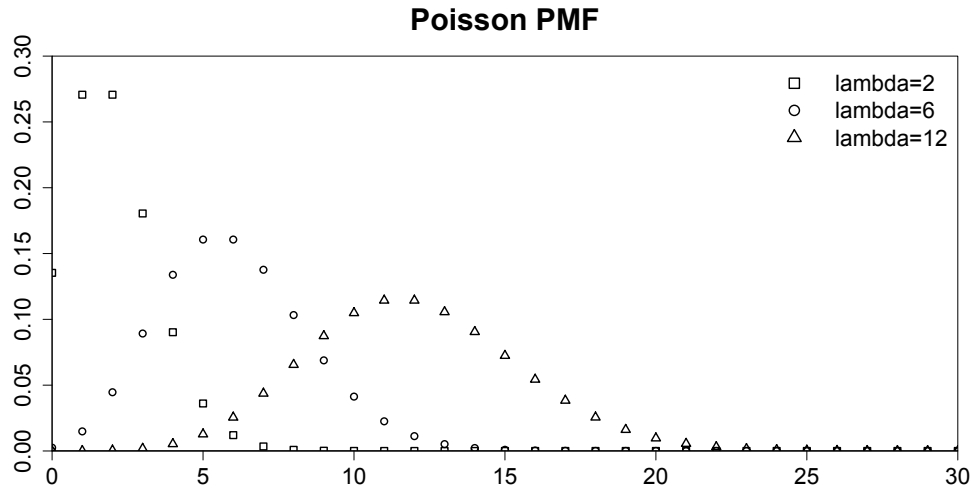


Figure 2.1: The probability mass function of the Poisson distribution as  $\lambda$  varies.

## 2.4 Claim Size Modeling

Claim severity modeling is about describing the variation in claim sizes. An insurance company's individual loss for a policy is not only non-negative, but can also be potentially very high. Probability distributions with heavy tails are often preferable, that is distributions which allow for occasional occurrences of very large values. The traditional approach is to assign a suitable family of probability distributions and estimate their parameters from historical claims  $z_1, \dots, z_n$ . Lack of historical data where it matters most financially is a challenge, and we should pay special attention to the tail which in some cases can be extreme.

### 2.4.1 Parametric Distributions

All sensible parametric models for claim sizes are of the form

$$Z = \beta Z_0, \quad (2.13)$$

where  $\beta > 0$  is known as a *parameter of scale* and  $Z_0$  is a standardized random variable corresponding to  $\beta = 1$ . By expressing  $Z$  on this form  $\beta$  can take up any external effects such as currency or inflation. Then the shape of the density function can remain as it was. The proportionality



is inherited by expectation and standard deviation, i.e.  $\xi_0$  and  $\sigma_0$ , for  $Z_0$ . Then the same quantities for  $Z$  are [Bølviken, 2014, p. 315],

$$\xi = \beta\xi_0, \quad \sigma = \beta\sigma_0. \quad (2.14)$$

### 2.4.1.1 The Gamma Distribution

The *Gamma* family is an important family for which the density function is

$$f(z) = \frac{(\alpha/\xi)^\alpha}{\Gamma(\alpha)} z^{\alpha-1} e^{-\alpha z/\xi}, \quad (2.15)$$

where  $\Gamma(\alpha) = \int_0^\infty z^{\alpha-1} e^{-z} dz$ ,  $\alpha$  is the shape parameter and  $\xi$  is the scale parameter.

The mean and the standard deviation of the Gamma variables are [Bølviken, 2014, p. 41]

$$E(Z) = \xi \quad \text{and} \quad \text{sd}(Z) = \frac{\xi}{\sqrt{\alpha}}. \quad (2.16)$$

We define the standard Gamma distribution as the Gamma distribution with mean one, i.e.  $\xi = 1$ , and shape  $\alpha$ . The density function of the standard Gamma simplifies to

$$f(z) = \frac{\alpha^\alpha}{\Gamma(\alpha)} z^{\alpha-1} e^{-\alpha z}, \quad z > 0 \quad \text{where} \quad \Gamma(\alpha) = \int_0^\infty z^{\alpha-1} e^{-z} dz. \quad (2.17)$$

Then  $Z \sim \text{Gamma}(\alpha, \xi)$  is obtained by  $Z = \xi Z_0$ , where  $Z_0 \sim \text{Gamma}(\alpha)$ .

From historical data  $z_1, \dots, z_n$  we can estimate the Gamma parameters  $\xi$  and  $\alpha$  by the maximum likelihood method. The likelihood of the density function of the standard Gamma is

$$L(\xi, \alpha) = \prod_{i=1}^n f(z_i | \xi, \alpha) = \left( \frac{(\alpha/\xi)^\alpha}{\Gamma(\alpha)} \right)^n \prod_{i=1}^n z_i^{\alpha-1} e^{-\alpha z_i}. \quad (2.18)$$

Then, the log-likelihood function is

$$\log L(\xi, \alpha) = n\alpha \log \left( \frac{\alpha}{\xi} \right) - n \log \{\Gamma(\alpha)\} + (\alpha - 1) \sum_{i=1}^n \log(z_i) - \frac{\alpha}{\xi} \sum_{i=1}^n z_i. \quad (2.19)$$

Differentiating  $\log L(\xi, \alpha)$  with respect to  $\xi$  and setting the result equal to zero, we obtain the maximum likelihood estimate of  $\xi$ ,

$$\frac{\partial \log L(\xi, \alpha)}{\partial \xi} = -\frac{n\alpha}{\xi} + \frac{\alpha}{\xi^2} \sum_{i=1}^n z_i = 0 \quad \Rightarrow \quad \hat{\xi} = \frac{1}{n} \sum_{i=1}^n z_i = \bar{z}. \quad (2.20)$$

This result can be inserted into the log-likelihood function, i.e.  $\log L(\bar{z}, \alpha)$ , such that we get a log-likelihood that only depends on  $\alpha$ ,

$$\begin{aligned} \log L(\alpha) &= n\alpha \log\left(\frac{\alpha}{\bar{z}}\right) - n \log\{\Gamma(\alpha)\} + (\alpha - 1) \sum_{i=1}^n \log(z_i) - \frac{\alpha}{\bar{z}} \sum_{i=1}^n z_i \\ &= n\alpha \left(\log\left(\frac{\alpha}{\bar{z}}\right) - 1\right) - n \log\{\Gamma(\alpha)\} + (\alpha - 1) \sum_{i=1}^n \log(z_i). \end{aligned} \quad (2.21)$$

If we differentiate  $\log L(\alpha)$  with respect to  $\alpha$  and setting the result equal to zero we will not obtain an explicit solution for  $\hat{\alpha}$ . Therefore the optimization with respect to  $\alpha$  must be done numerically in R.

In R, we can generate Gamma variables from the computer command  $Z = \xi * \text{rgamma}(m, \alpha, \alpha)$ . Here the Gamma variables are defined as  $Z = \xi G$  where  $G$  has mean 1 and shape  $\alpha$ .

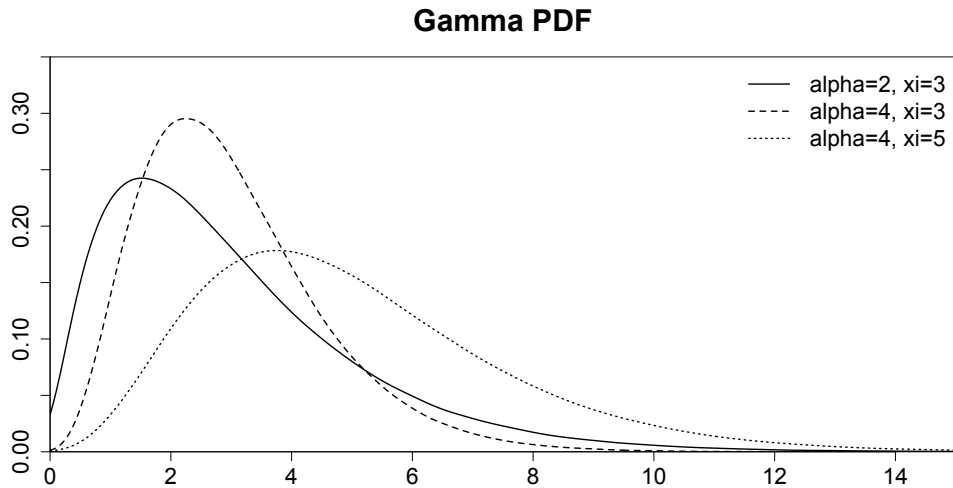


Figure 2.2: The pdf of the Gamma distribution as  $\alpha$  and  $\xi$  vary.

In Figure 2.2 the Gamma pdf for several pairs of  $\alpha$  and  $\xi$  is plotted. With fixed  $\alpha$ , the graph is more right skewed as  $\xi$  becomes larger. With fixed  $\xi$ , the Gamma pdf is less spread-out as  $\alpha$  increases.

### 2.4.1.2 The Weibull Distribution

The Weibull distribution is a widely used distribution because of its versatility. The density function is

$$f(z) = \frac{\alpha}{\beta} \left( \frac{z}{\beta} \right)^{\alpha-1} e^{-(z/\beta)^\alpha}, \quad z > 0. \quad (2.22)$$

The mean and the standard deviation are [Bølviken, 2014, p. 43]

$$E(Z) = \beta\Gamma(1 + 1/\alpha) \quad \text{and} \quad \text{sd}(Z) = \beta\sqrt{\Gamma(1 + 2/\alpha) - \Gamma(1 + 1/\alpha)^2}. \quad (2.23)$$

Estimates of the parameters  $\alpha$  and  $\beta$  can be obtained by the maximum likelihood method. The likelihood function is

$$\begin{aligned} L(\alpha, \beta) &= \prod_{i=1}^n f(z_i | \alpha, \beta) \\ &= \frac{\alpha^n}{\beta^{n\alpha}} e^{-\sum_{i=1}^n (z_i/\beta)^\alpha} \prod_{i=1}^n z_i^{\alpha-1}. \end{aligned} \quad (2.24)$$

The log-likelihood function is

$$\log L(\alpha, \beta) = n \log(\alpha) - n\alpha \log(\beta) + (\alpha - 1) \sum_{i=1}^n \log(z_i) - \frac{1}{\beta^\alpha} \sum_{i=1}^n z_i^\alpha. \quad (2.25)$$

Differentiating  $\log L(\alpha, \beta)$  with respect to  $\beta$  and setting the result equal to zero we obtain the maximum likelihood estimate of  $\beta$ ,

$$\frac{\partial \log L(\alpha, \beta)}{\partial \beta} = -\frac{n\alpha}{\beta} + \frac{\alpha}{\beta^{\alpha+1}} \sum_{i=1}^n z_i^\alpha = 0 \quad \Rightarrow \quad \hat{\beta}_\alpha = \left( \frac{1}{n} \sum_{i=1}^n z_i^\alpha \right)^{1/\alpha}. \quad (2.26)$$

This result can be inserted into the log-likelihood function in (2.25) such that we get a log-likelihood that only depends on  $\alpha$ ,

$$\begin{aligned}
\log L(\alpha) &= n \log(\alpha) - n\alpha \log \left( \left( \frac{1}{n} \sum_{i=1}^n z_i^\alpha \right)^{1/\alpha} \right) + (\alpha - 1) \sum_{i=1}^n \log(z_i) \\
&\quad - \frac{1}{\left( \left( \frac{1}{n} \sum_{i=1}^n z_i^\alpha \right)^{1/\alpha} \right)^\alpha} \sum_{i=1}^n z_i^\alpha \\
&= n[\log(\alpha) + \log(n) - \log \left( \sum_{i=1}^n z_i^\alpha \right) - 1] + (\alpha - 1) \sum_{i=1}^n \log(z_i),
\end{aligned} \tag{2.27}$$

where the optimization with respect to  $\alpha$  can be done numerically in R. Furthermore, the computer commands for generating Weibull variables are  $Z = \text{rweibull}(m, \alpha, \beta)$ .

In Figure 2.3 the Weibull pdf for several pairs of  $\alpha$  and  $\beta$  is plotted. The Weibull density extends over the positive axis and is skewed to the right when  $\alpha < 3.3$ , to the left when  $\alpha > 3.3$  and when  $\alpha = 3.3$  it is similar to the normal distribution.

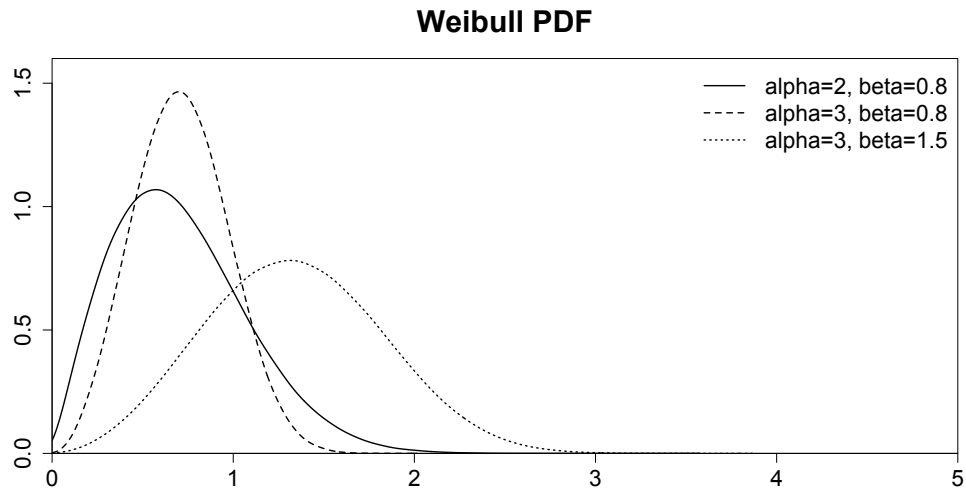


Figure 2.3: The pdf of the Weibull distribution as  $\alpha$  and  $\beta$  vary.

### 2.4.1.3 The Pareto Distribution

The Pareto distribution is among the most heavy-tailed of all models in practical use and is essential for modeling extreme losses, especially in the

more risky types of insurance. Hence it is a conservative choice when modeling the claim size. Generally, the density function of the Pareto distribution is

$$g(x) = \frac{\alpha x_m^\alpha}{x^{1+\alpha}}, \quad x \geq x_m, \quad (2.28)$$

where  $x_m$  is the (necessarily positive) minimum possible value of  $X$ , and  $\alpha$  is a positive parameter. We assume that the smallest possible value of  $x$  is 1, and the density is

$$g(x) = \frac{\alpha}{x^{1+\alpha}}, \quad x \geq 1. \quad (2.29)$$

In order to make the density support values from zero we let  $Z = \beta(x-1)$  such that  $x(z) = 1 + (z/\beta)$  by inversion. The probability density function of  $Z$  is

$$f(z) = g(x(z)) \left| \frac{\partial x(z)}{\partial z} \right| = \frac{\alpha/\beta}{(1 + z/\beta)^{1+\alpha}}, \quad z > 0. \quad (2.30)$$

Here  $\alpha > 0$  is the shape parameter and  $\beta > 0$  is the scale parameter. This distribution is often called the Lomax distribution and is essentially a Pareto distribution that has been shifted so that its support begins at zero.

The mean and the standard deviation of Pareto variables are [Bølviken, 2014, p. 43]

$$E(Z) = \frac{\beta}{\alpha - 1} \quad \text{and} \quad \text{sd}(Z) = E(Z) \sqrt{\frac{\alpha}{\alpha - 2}}, \quad (2.31)$$

where  $\alpha > 1$  and  $\alpha > 2$  for the mean and standard deviation to be defined, respectively.

From historical data  $z_1, \dots, z_n$  we can estimate the Pareto parameters  $\alpha$  and  $\beta$  by the maximum likelihood method. The likelihood of the density function is

$$L(\alpha, \beta) = \prod_{i=1}^n f(z_i | \alpha, \beta) = \frac{(\alpha/\beta)^n}{\prod_{i=1}^n (1 + z_i/\beta)^{1+\alpha}}. \quad (2.32)$$

The log-likelihood function is

$$\log L(\alpha, \beta) = n \log(\alpha) - n \log(\beta) - (1 + \alpha) \sum_{i=1}^n \log \left( 1 + \frac{z_i}{\beta} \right). \quad (2.33)$$

Differentiating with respect to  $\alpha$ , we obtain the maximum likelihood estimate,

$$\frac{\partial \log L(\alpha, \beta)}{\partial \alpha} = \frac{n}{\alpha} - \sum_{i=1}^n \log \left( 1 + \frac{z_i}{\beta} \right) = 0,$$

which gives

$$\hat{\alpha}_\beta = \frac{n}{\sum_{i=1}^n \log(1 + z_i/\beta)}. \quad (2.34)$$

This can be inserted into the log-likelihood function in 2.33 such that we get a log-likelihood that only depends on  $\beta$ ,

$$\begin{aligned} \log L(\beta) &= n \log \left( \frac{n}{\sum_{i=1}^n \log(1 + z_i/\beta)} \right) - n \log(\beta) \\ &\quad - \left( 1 + \frac{n}{\sum_{i=1}^n \log(1 + z_i/\beta)} \right) \sum_{i=1}^n \log \left( 1 + \frac{z_i}{\beta} \right) \\ &= n \left[ \log(n) - \log \left( \sum_{i=1}^n \log \left( 1 + \frac{z_i}{\beta} \right) \right) - \log(\beta) - 1 \right] \\ &\quad - \sum_{i=1}^n \log \left( 1 + \frac{z_i}{\beta} \right). \end{aligned} \quad (2.35)$$

The optimization with respect to  $\beta$  can be done numerically in R.

We can use the inverse of the Pareto distribution function in Algorithm 1 as a Pareto sampler. This is given more thoroughly in Appendix A.2.

---

**Algorithm 1** Pareto generator

---

- 1: Input:  $\alpha, \beta$
  - 2: Generate  $U^* \sim \text{uniform}$
  - 3: Return  $X^* \leftarrow \beta((U^*)^{-1/\alpha} - 1)$
- 

Figure 2.4 illustrates the pdf of the Pareto distribution when  $\alpha$  and  $\beta$  vary. When  $\alpha$  is fixed and  $\beta$  increases, the pdf becomes less right skewed. However, when  $\beta$  is fixed and  $\alpha$  increases, the pdf becomes more right skewed.

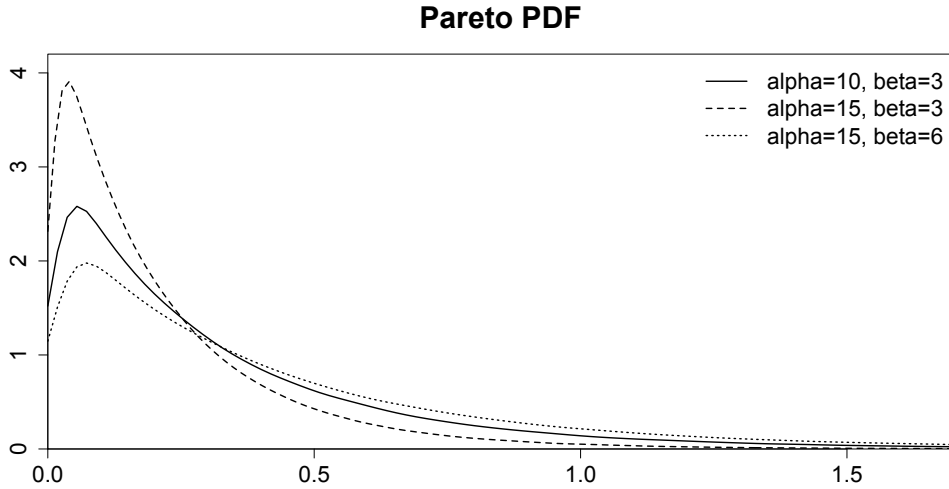


Figure 2.4: The pdf of the Pareto distribution as  $\alpha$  and  $\beta$  vary.

#### 2.4.1.4 The Extended Pareto Distribution

The extended Pareto distribution is a generalization of the ordinary Pareto distribution with density function

$$f(z) = \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \frac{1}{\beta} \frac{(z/\beta)^{\theta-1}}{(1 + z/\beta)^{\alpha+\theta}}, \quad z > 0, \quad (2.36)$$

where  $\alpha, \beta, \theta > 0$ . In the extended Pareto distribution there is an additional parameter,  $\theta$ , which creates useful flexibility for the shape. The extended Pareto distribution is often called the Beta prime distribution.

The mean and the standard deviation of the extended Pareto distribution are [Bølviken, 2014, p. 324]

$$E(Z) = \frac{\theta\beta}{\alpha - 1} \quad \text{and} \quad \text{sd}(Z) = E(Z) \sqrt{\frac{\alpha + \theta - 1}{\theta(\alpha - 2)}}, \quad (2.37)$$

for  $\alpha > 1$  and  $\alpha > 2$ , respectively. This distribution reduces to the ordinary Pareto distribution when  $\theta = 1$ .

Sampling from the extended Pareto distribution can be complicated, but this can be simplified by utilizing that the extended Pareto variable,  $Z$ , with parameters  $(\alpha, \beta, \theta)$  can be written as

$$Z = \frac{\theta\beta}{\alpha} \frac{G_\theta}{G_\alpha}, \quad (2.38)$$

where  $G_\theta \sim \text{Gamma}(\theta)$ ,  $G_\alpha \sim \text{Gamma}(\alpha)$  are two independent Gamma variables with mean 1. The representation is proven in Appendix A.3.

By the maximum likelihood method we can estimate the parameters  $\alpha$ ,  $\theta$  and  $\beta$  of the extended Pareto from the historical losses  $z_1, \dots, z_n$ . The likelihood is

$$\begin{aligned} L(\alpha, \theta, \beta) &= \prod_{i=1}^n f(z_i | \alpha, \theta, \beta) \\ &= \frac{\Gamma(\alpha + \theta)^n}{\Gamma(\alpha)^n \Gamma(\theta)^n} \frac{1}{\beta^n} \frac{\prod_{i=1}^n (z_i/\beta)^{\theta-1}}{\prod_{i=1}^n (1 + z_i/\beta)^{\alpha+\theta}} \\ &= \frac{\Gamma(\alpha + \theta)^n}{\Gamma(\alpha)^n \Gamma(\theta)^n} \frac{1}{\beta^{n\theta}} \frac{\prod_{i=1}^n z_i^{\theta-1}}{\prod_{i=1}^n (1 + z_i/\beta)^{\alpha+\theta}}. \end{aligned} \quad (2.39)$$

The log-likelihood function is

$$\begin{aligned} \log L(\alpha, \theta, \beta) &= n[\log(\Gamma(\alpha + \theta)) - \log(\Gamma(\alpha)) - \log(\Gamma(\theta)) - \theta \log(\beta)] \\ &\quad + (\theta - 1) \sum_{i=1}^n \log(z_i) - (\alpha + \theta) \sum_{i=1}^n \log(1 + z_i/\beta). \end{aligned} \quad (2.40)$$

The optimization of the parameters  $\alpha$ ,  $\theta$  and  $\beta$  can be done numerically in R by the `optim()` function. The computer commands for generating extended Pareto variables are `Z = beta * theta / alpha * rgamma(m, theta, theta) / rgamma(m, alpha, alpha)`.

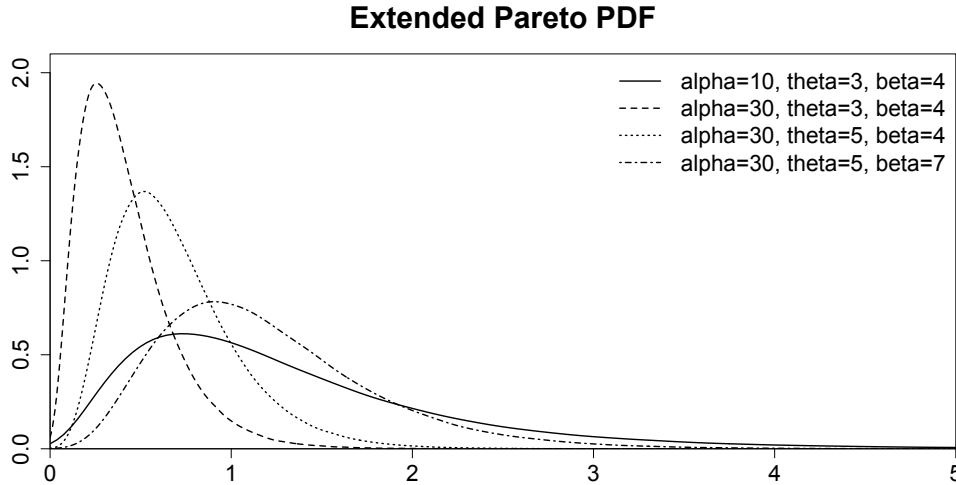


Figure 2.5: The pdf of the extended Pareto distribution as  $\alpha$ ,  $\theta$  and  $\beta$  vary.



In Figure 2.5 the pdf of the extended Pareto distribution for several values of  $\alpha$ ,  $\theta$  and  $\beta$  is plotted. With fixed  $\theta$  and  $\beta$ , the graph is slightly less right skewed as  $\alpha$  becomes larger. With fixed  $\alpha$  and  $\beta$ , an increase in  $\theta$  yields a less flat graph. As  $\beta$  increases with fixed  $\alpha$  and  $\theta$ , there is an increase in the degree of right skewness.

#### 2.4.1.5 4-parameter Distribution

Assume that

$$Z = \beta X^\eta, \quad \text{where} \quad X = \frac{G_\theta}{G_\alpha}. \quad (2.41)$$

Here  $Z$  is a positive variable with  $\beta > 0$  and  $\eta > 0$ , and  $G_\theta$  and  $G_\alpha$  are standard Gamma distributions with mean one and shape  $\theta > 0$  and  $\alpha > 0$ , respectively.

By equation (2.38) we know that the variable  $X$  can be written as an extended Pareto variable with  $\beta = \alpha/\theta$  and pdf

$$\begin{aligned} g(x) &= \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \frac{\theta}{\alpha} \frac{(\theta x/\alpha)^{\theta-1}}{(1 + \theta x/\alpha)^{\alpha+\theta}} \\ &= \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \frac{\theta}{\alpha} \frac{(\theta x/\alpha)^{\theta-1} (\alpha/\theta)^{\alpha+\theta}}{(1 + \theta x/\alpha)^{\alpha+\theta} (\alpha/\theta)^{\alpha+\theta}} \\ &= \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \left(\frac{\alpha}{\theta}\right)^\alpha \frac{x^{\theta-1}}{(\alpha/\theta + x)^{\alpha+\theta}}, \quad x > 0. \end{aligned} \quad (2.42)$$

By inversion, we get from  $Z = \beta X^\eta$ , that  $x(z) = (z/\beta)^{\frac{1}{\eta}}$ . Moreover, the probability density function of  $Z$  is given by

$$\begin{aligned} f(z) &= g(x(z)) \left| \frac{\partial x(z)}{\partial z} \right| \\ &= \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \left(\frac{\alpha}{\theta}\right)^\alpha \frac{((z/\beta)^{\frac{1}{\eta}})^{\theta-1} (z/\beta)^{\frac{1}{\eta}}}{(\alpha/\theta + (z/\beta)^{\frac{1}{\eta}})^{\alpha+\theta} z\eta} \\ &= \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \left(\frac{\alpha}{\theta}\right)^\alpha \frac{(z/\beta)^{\frac{\theta}{\eta}} (\theta/\alpha)^{\alpha+\theta}}{z\eta (\alpha/\theta + (z/\beta)^{\frac{1}{\eta}})^{\alpha+\theta} (\theta/\alpha)^{\alpha+\theta}} \\ &= \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \left(\frac{\theta}{\alpha}\right)^\theta \frac{(z/\beta)^{\frac{\theta}{\eta}}}{z\eta (1 + (\theta/\alpha)(z/\beta)^{\frac{1}{\eta}})^{\alpha+\theta}}, \quad z > 0. \end{aligned} \quad (2.43)$$

By the result in [Bølviken, 2014, p. 334] we know that the expectation for the extended Pareto random variable  $X$  is

$$E(X^i) = \left(\frac{\alpha}{\theta}\right)^i \frac{\Gamma(\alpha - i)\Gamma(\theta + i)}{\Gamma(\alpha)\Gamma(\theta)}, \quad (2.44)$$

Then for  $Z = \beta X^\eta$  we get the expectation

$$E(Z) = E(\beta X^\eta) = \beta E(X^\eta) = \beta \left(\frac{\alpha}{\theta}\right)^\eta \frac{\Gamma(\alpha - \eta)\Gamma(\theta + \eta)}{\Gamma(\alpha)\Gamma(\theta)}. \quad (2.45)$$

For the standard deviation we use  $\text{var}(Z) = E(Z^2) - (E(Z))^2$  and simplify. This gives

$$\begin{aligned} \text{sd}(Z) &= \sqrt{\beta^2 \left(\frac{\alpha}{\theta}\right)^{2\eta} \frac{\Gamma(\alpha - 2\eta)\Gamma(\theta + 2\eta)}{\Gamma(\alpha)\Gamma(\theta)} - \left(\beta \left(\frac{\alpha}{\theta}\right)^\eta \frac{\Gamma(\alpha - \eta)\Gamma(\theta + \eta)}{\Gamma(\alpha)\Gamma(\theta)}\right)^2} \\ &= \beta \left(\frac{\alpha}{\theta}\right)^\eta \sqrt{\frac{\Gamma(\alpha - 2\eta)\Gamma(\theta + 2\eta)}{\Gamma(\alpha)\Gamma(\theta)} - \left(\frac{\Gamma(\alpha - \eta)\Gamma(\theta + \eta)}{\Gamma(\alpha)\Gamma(\theta)}\right)^2}. \end{aligned} \quad (2.46)$$

By the maximum likelihood method we can estimate the parameters  $\alpha$ ,  $\theta$ ,  $\beta$  and  $\eta$  of the 4-parameter from the historical losses  $z_1, \dots, z_n$ . The likelihood function is

$$\begin{aligned} L(\alpha, \theta, \beta, \eta) &= \prod_{i=1}^n f(z_i | \alpha, \theta, \beta, \eta) \\ &= \frac{\Gamma(\alpha + \theta)^n}{\Gamma(\alpha)^n \Gamma(\theta)^n} \left(\frac{\theta}{\alpha}\right)^{n\theta} \frac{\prod_{i=1}^n (z_i/\beta)^{\frac{\theta}{\eta}}}{\prod_{i=1}^n z_i \eta (1 + (\theta/\alpha)(z_i/\beta)^{\frac{1}{\eta}})^{\alpha+\theta}}. \end{aligned} \quad (2.47)$$

The log-likelihood function is

$$\begin{aligned} \log L(\alpha, \theta, \beta, \eta) &= n[\log(\Gamma(\alpha + \theta)) - \log(\Gamma(\alpha)) - \log(\Gamma(\theta))] \\ &\quad + \theta \log(\theta) - \theta \log(\alpha) - \frac{\theta}{\eta} \log(\beta) - \log(\eta)] \\ &\quad + \left(\frac{\theta}{\eta} - 1\right) \sum_{i=1}^n \log(z_i) - (\alpha + \theta) \sum_{i=1}^n \log\left(1 + \frac{\theta}{\alpha} \left(\frac{z_i}{\beta}\right)^{\frac{1}{\eta}}\right). \end{aligned} \quad (2.48)$$

Again, the optimization of the parameters  $\alpha$ ,  $\theta$ ,  $\beta$  and  $\eta$  can be done numerically in R by the `optim()` function. Furthermore, the computer command in order to generate samples from the 4-parameter distribution is  $Z = \beta * (\text{rgamma}(m, \theta, \theta) / \text{rgamma}(m, \alpha, \alpha))^{\eta}$ .

In Figure 2.6 the pdf of the 4 parameter distribution for several values of  $\alpha$ ,  $\theta$ ,  $\beta$  and  $\eta$  is plotted. With fixed  $\theta$ ,  $\beta$  and  $\eta$ , the graph is slightly less right skewed as  $\alpha$  becomes larger. With fixed  $\alpha$ ,  $\beta$  and  $\eta$ , an increase in  $\theta$  yields a less flat graph. With fixed  $\alpha$ ,  $\theta$  and  $\eta$ , the graph is more right skewed as  $\beta$  becomes larger. As  $\eta$  increases with fixed  $\alpha$ ,  $\theta$  and  $\beta$ , the graph becomes more spread-out.

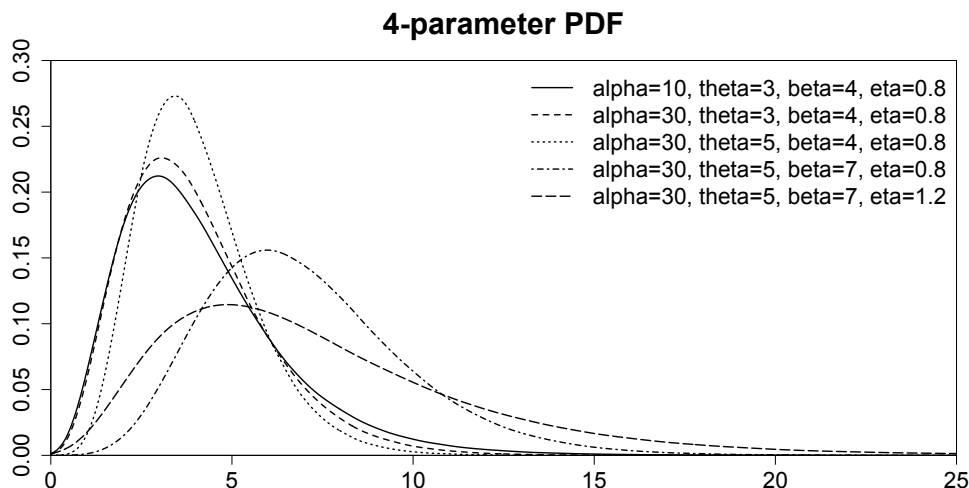


Figure 2.6: The pdf of the 4 parameter distribution as  $\alpha$ ,  $\theta$ ,  $\beta$  and  $\eta$  vary.

### 2.4.2 Non-parametric Model

Claim sizes can also be modeled non-parametrically, where each claim  $z_i$  of the past is assigned a probability  $1/n$  of reappearing in the future. Each claim is regarded as a random variable  $\hat{Z}$  for which

$$\Pr(\hat{Z} = z_i) = \frac{1}{n}, \quad i = 1, \dots, n. \quad (2.49)$$

Then the cumulative distribution function is a step function that jumps  $1/n$  at each of the  $n$  data points. Note that these weights are positive and sum to 1, and the distribution is called the *empirical distribution*. The mean and standard deviation of the empirical distribution are

$$E(\hat{Z}) = \sum_{i=1}^n \frac{1}{n} z_i = \bar{z} \quad \text{and} \quad \text{sd}(\hat{Z}) = \sqrt{\sum_{i=1}^n \frac{1}{n} (z_i - \bar{z})^2}. \quad (2.50)$$

The empirical distribution is easy to set up and to simulate from, and we only need to assume that the historical data are independent and identically distributed. However, skewness tends to be small, and no simulated claim can be larger than what we know from the historical data. Hence, we need a large number of observations in order for the empirical distribution to be a good choice.

## 2.5 Reserve

The reserve (solvency capital) is the funds an insurance company is obligated to set aside to cover future obligations. Hence, the reserve ensures regulators that the insurance company is able to cover significant losses [Bølviken, 2014, p. 5]. The mathematical formulation is

$$\Pr(\mathcal{X} > q_\epsilon) = \epsilon \quad (2.51)$$

where  $\mathcal{X}$  is the the total claim at portfolio level and  $\epsilon$  is a small number (for example 1%). Then the reserve,  $q_\epsilon$ , should be large enough such that the probability that the total loss  $\mathcal{X}$  exceeds the amount of the reserve is very small.

### 2.5.1 Monte Carlo Simulation

Calculating the reserve is essential in general insurance and the *Monte Carlo* method is the general tool for the job since the underlying processes can be too complex for analytic manipulation. The technique was first developed by three scientists who worked on a nuclear weapon project called the Manhattan Project. They named the method after the gambling hot spot in Monaco, since chance and random outcomes are central to the modeling technique, similar to games like roulette, dice and slot machines [Thomopoulos, 2012, p. 1].

The Monte Carlo method rely on repeated random sampling to study properties of a statistic's sampling distribution and its behavior. The results of these numerous scenarios can give us a "most likely" case , along with a statistical distribution to understand the risk and uncertainty involved.

In actuarial applications the Monte Carlo method is used to estimate an upper percentile of the loss distribution.

**Theorem 2.5.1. Strong Law of Large Numbers.**

Let  $X_1, \dots, X_N$  be a sequence of independent and identically distributed random variables. Assume that  $E(X_1) < \infty$ . For  $N \geq 1$ , denote the empirical mean of  $X_1, \dots, X_N$  by

$$\bar{X}_N := \frac{1}{N} \sum_{i=1}^N X_i. \quad (2.52)$$

Then, the Strong Law of Large Numbers (SLLN) states that [Graham and Talay, 2013, p. 13]

$$\lim_{N \rightarrow \infty} \bar{X}_N = E(X_1), \quad P - a.s. \quad (2.53)$$

The accuracy of the estimate depends on the number of simulation used in the Monte Carlo method. Hence, the Strong Law of Large Numbers is at the core of the Monte Carlo method used below.

## 2.5.2 Estimation of the Reserve

Suppose we use Monte Carlo simulation to generate a sample of  $m = 1\,000\,000$  values of the total loss  $\mathcal{X}$ , and order them from smallest to largest. With solvency levels of  $\epsilon_1 = 0.05$  and  $\epsilon_2 = 0.01$ , we are interested in the upper percentiles  $q_{0.95}$  and  $q_{0.99}$  of the portfolio liability  $\mathcal{X}$ . A Monte Carlo simulation with  $m = 1\,000\,000$  should give a good estimate of the true underlying distribution of  $\mathcal{X}^*$ .

First, we draw  $\mathcal{X}_1^*, \dots, \mathcal{X}_m^*$  and sort them in rising order as  $\mathcal{X}_{(1)}^* \leq \dots \leq \mathcal{X}_{(m)}^*$ . Then, we let  $\hat{q}_\epsilon^* = \mathcal{X}_{((1-\epsilon)m)}^*$ . By Theorem 2.5.1 it can be shown that  $\hat{q}_\epsilon^* \xrightarrow{m \rightarrow \infty} \hat{q}_\epsilon$ . A more detailed procedure is given in the algorithm below.

---

**Algorithm 2** Computing the reserve with Monte Carlo
 

---

- 1: Input:  $m, J\mu T, \hat{\xi}, \hat{\alpha}, \epsilon$
  - 2:  $\mathcal{X}^* \leftarrow 0$
  - 3: **for**  $j = 1, \dots, m$  **do**
  - 4: Generate  $\mathcal{N}_j^* \sim \text{Poisson}(J\mu T)$
  - 5: **for**  $i = 1, \dots, \mathcal{N}_j^*$  **do**
  - 6: Draw  $Z_i^* \sim \text{Gamma}(\hat{\alpha}, \hat{\xi})$
  - 7:  $\mathcal{X}_j^* \leftarrow Z_1^*, \dots, Z_{\mathcal{N}_j^*}^*$
  - 8: Sort  $\mathcal{X}_1^*, \dots, \mathcal{X}_m^*$  as  $\mathcal{X}_{(1)}^* \leq \dots \leq \mathcal{X}_{(m)}^*$
  - 9: Return  $\hat{q}_\epsilon^* \leftarrow \mathcal{X}_{((1-\epsilon)m)}^*$
- 

In Algorithm 2 we have used a Poisson/Gamma portfolio where  $J\mu T$  is the expected number of claims and  $m$  is the number of simulations used. The parameters  $\hat{\alpha}$  and  $\hat{\xi}$  are estimated from the maximum likelihood method. Line 1 and 6 can be replaced by another set of estimated parameters and claim size distribution, respectively.

# Chapter 3

## Simulation

Today computers have sufficient power to run complex simulations that allow for the analysis of models that are not suitable for analytic approaches. We want to compare the reserve from the different true distributions using measures such as bias and root mean squared error, with special attention to the performance of the 3- and 4-parameter distributions relative to the two-parameter distributions. In this way we can determine if the 3- and 4-parameter models provide a good fit when the sample are from a two-parameter distribution. To assess this we will perform a simulation study.

### 3.1 Plan for the Simulation Study

The simulation involves specific steps in order for the simulation study to be successful. These can be summarized in three steps, where the goal is to determine values relating to the distribution of the aggregated loss  $\mathcal{X}$  [Panjer et al., 2008, p. 612].

1. Build a model for  $\mathcal{X}$  which depends on random the variables  $\mathcal{N}$  and  $Z_1, Z_2, \dots, Z_N$  with known distributions and independencies.
2. For  $j = 1, \dots, m$  generate random values for  $\mathcal{N}$  and  $Z_1, Z_2, \dots, Z_N$  using the model from step 1, and then compute  $\hat{\mathcal{X}}_j^*$ .
3. Estimate quantiles of interest which in our case is the upper percentiles  $\hat{q}_{0.95}^*$  and  $\hat{q}_{0.99}^*$  as described in Algorithm 2.

## 3.2 The model

The model used to compute the aggregated loss, and furthermore the reserve, is a combination of the model for the number of claims  $\mathcal{N}$  and claim sizes  $\mathbf{Z} = Z_1, \dots, Z_N$ . We will only use the Poisson model for modeling claim frequency. For the claim sizes, however, we will use either the Gamma, Weibull, Pareto, extended Pareto or the 4-parameter distribution as the true distribution, and thereafter estimate the parameters for all five distributions and compute the corresponding reserve. We rerun the simulation for each of the five distributions as the true distribution. Hence, for the steps in section 3.1, we will

- Try three different sample sizes.
- Use each of the five distributions as the true distribution with suitable parameters.

For each scenario we will first simulate a certain number of samples  $n$  from the true distribution. Thereafter the parameters of each of the five distributions will be estimated using maximum likelihood, where the starting values are determined by the method of moments. Finally we compute the corresponding reserves, with the estimated parameters, by  $m$  Monte Carlo simulations as specified in Algorithm 2.

Hence, if we use Gamma as the true distribution, the entire process becomes

$$n \text{ claims } z_1, \dots, z_n \longrightarrow \hat{\mu}, \hat{\alpha}, \hat{\xi} \longrightarrow \hat{\mathcal{X}}_1^*, \dots, \hat{\mathcal{X}}_m^* \longrightarrow \hat{q}_{(1-\epsilon)m}^* = \hat{\mathcal{X}}_{(1-\epsilon)m}^*$$

where the R functions `parEstimate_gamma()` and `reserve_gamma()` estimate the parameters and reserve, respectively (see Appendix B.2). That is, if we use Gamma as the true distribution we want to compute the reserve for all the five distributions. First we need to sample  $n$  random variables by the R-command `Z = xi * rgamma(m, alpha, alpha)`. Then the functions `parEstimate_gamma()`, ..., `parEstimate_fourPar()` take the  $\mathbf{Z}$  values as input and estimate the parameters for each of the distributions by the maximum likelihood method. For the extended Pareto and 4-parameter distributions we use the method of moments estimates as starting values for the `optim()` function. The maximum likelihood estimated parameters are sent to some other functions, `reserve_gamma()`, ..., `reserve_fourPar()`, which calculate the corresponding reserves as stated in Algorithm 2. This is repeated  $N$  times for each distribution as the true distribution and for three different sample sizes  $n$ .



In order to generate continuous random variables from the distributions we will use the following R-commands.

Table 3.1: Generating continuous random variables

Distribution	Random variables
Gamma	$Z = \xi * \text{rgamma}(m, \alpha, \alpha)$
Weibull	$Z = \text{rweibull}(m, \alpha, \beta)$
Pareto	$Z = \beta((U)^{-1/\alpha} - 1), U \sim \text{uniform}$
extended Pareto	$Z = \beta * \theta / \alpha * \text{rgamma}(m, \theta, \theta) / \text{rgamma}(m, \alpha, \alpha)$
4 Parameter	$Z = \beta * (\text{rgamma}(m, \theta, \theta) / \text{rgamma}(m, \alpha, \alpha))^{\eta}$

### 3.3 Parameter Setting

For each of the five distributions, we need to find suitable parameter values. The claim size distributions we are considering are skewed to the right, non-negative and unimodal. The Gamma( $\alpha, \xi$ ) and Weibull( $\alpha, \beta$ ) distributions can be used when the tail of the probability density function is not *too heavy*. On the other hand, the Pareto( $\alpha, \beta$ ) distribution is very heavy-tailed and is a popular choice in branches with high probability of large claims. To gain more flexibility we can add parameters to the models. The extended Pareto( $\alpha, \theta, \beta$ ) and 4-parameter( $\alpha, \theta, \beta, \eta$ ) are such models, with three and four parameters, respectively. Keeping these characteristics in mind, we can find suitable parameter values presented in the table and plot below. For simplicity reasons we will choose the parameters such that the mean of the distributions is close to one. This can easily be scaled up to any desired expectation.

Table 3.2: Parameter values

Distribution	Parameter values
Gamma	$\alpha = 3.4, \xi = 1$
Weibull	$\alpha = 3.4, \beta = 1.1$
Pareto	$\alpha = 3.4, \beta = 2.4$
extended Pareto	$\alpha = 3.4, \theta = 2.0, \beta = 1.2$
4 parameter	$\alpha = 3.4, \theta = 2.9, \beta = 0.6, \eta = 1.2$

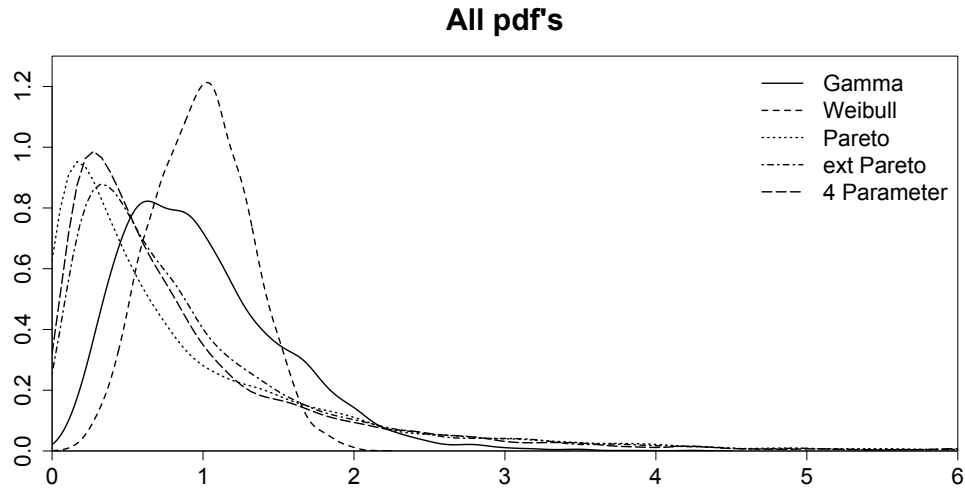


Figure 3.1: Probability density function for all five distributions used in the simulation study

Moreover, the following settings will be used.

- Number of simulations in each experiment:  $N = 500$
- Sample sizes:  $n = 5\,000, 500, 50$
- Expected number of occurrences:  $J\mu T = 50$
- Number of MC-simulations:  $m = 1\,000\,000$

We know that when the sample size decreases, the uncertainty in the estimate of the reserve becomes larger. Despite this, we want to know if it is still possible to use the 3- and 4-parameter distributions for all sample sizes. Therefore we use three different sample sizes. The Monte Carlo method will use  $m = 1\,000\,000$  simulations to provide an accurate assessment of the total risk. The simulation models specified in the R-code in Appendix B.2 are now ready to be executed.

# Chapter 4

## Reserve

In this chapter, the reserves  $q_{0.95}$  and  $q_{0.99}$  will be estimated for each of the five distributions. When using more than one model for computing the reserve, we are faced with the task of choosing between the models. Quantities such as the *bias* and *root mean squared error* can help us assess the fit of these models. We will present the bias and root mean squared error (RMSE) for each of the five distributions and evaluate their fit relative to the true distribution.

### 4.1 Error Analysis

The bias of an estimator is the difference between the estimator's expected value and the true value of the parameter being estimated. An estimator with bias equal to zero is called *unbiased* [Casella and Berger, 2002, p. 330]. We can calculate the bias of the reserve from

$$\hat{b} = E(\hat{q}_\epsilon - q_\epsilon) \simeq \frac{1}{N} \sum_{j=1}^N (\hat{q}_{\epsilon,j}^* - q_\epsilon^*), \quad (4.1)$$

where  $\hat{q}_{\epsilon,j}^*$  is the estimated reserve and  $q_\epsilon^*$  is the true reserve estimated with MC methods. A negative bias indicates that the estimated reserve is *underestimated* and a positive bias indicates that we have an *overestimated* reserve.

The root mean squared error is a commonly used measures of the differences between the sample and population values estimated by a model. The value is computed by taking the average of the squared differences between each estimated value and its corresponding true value [Ranka et al., 2010, p. 430]. Hence, the RMSE is the standard deviation of the residuals (prediction errors). The RMSE is simply the square root of the mean squared

error, and therefore the RMSE amplifies and severely punishes outliers. The RMSE for the reserve can be calculated from

$$\widehat{\text{RMSE}} = \sqrt{E(\hat{q}_\epsilon - q_\epsilon)^2} \simeq \sqrt{\frac{1}{N} \sum_{j=1}^N (\hat{q}_{\epsilon,j}^* - q_\epsilon^*)^2}. \quad (4.2)$$

The true reserves,  $q_\epsilon^*$ , are calculated by Algorithm 2 with the true parameter values specified in Table 3.2, where we use  $m = 10\,000\,000$  MC-simulations. Using ten million simulations will make Monte Carlo uncertainty very small indeed [Bølviken, 2014, p. 356]. The true reserves are given in Table 4.1 below.

Table 4.1: True reserve for  $m = 10\,000\,000$  MC-simulations

	Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	63.670	61.825	72.983	69.788	72.997
$\epsilon = 0.01$	69.852	67.348	87.431	81.430	89.714

#### 4.1.1 Sample Size: 5 000

First, we check the sample size  $n = 5\,000$ . From the *Law of Large Numbers*, in Theorem 2.5.1, we know that as the number of observations becomes large, the variance approaches zero. When the sample size is  $n = 5\,000$  the maximum likelihood method tends to produce more or less the same estimated parameters for each run  $j = 1, \dots, N$ , which again produces more precise reserve estimates.

Table 4.2: True distribution: Gamma ( $n = 5\,000$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-0.011	0.119	14.222	3.965	1.143
	$\widehat{\text{RMSE}}$	0.502	0.515	14.288	7.197	4.106
$\epsilon = 0.01$	$\hat{b}$	-0.012	0.087	20.188	6.202	1.449
	$\widehat{\text{RMSE}}$	0.553	0.556	20.286	11.209	4.741

When Gamma is the true distribution we get the biases and root mean squared errors as presented in Table 4.2. The Gamma distribution obviously has the best fit with lowest bias and RMSE for both the  $q_{0.95}$  and  $q_{0.99}$  reserves. The resembling Weibull distribution is not far behind, providing a good fit to the Gamma distributed data. The Pareto distribution, however, fits poorly. This is as expected given that the Pareto distribution is a very heavy-tailed distribution and the Gamma distribution is a moderate-tailed distribution. Furthermore, the extended Pareto distribution, with three parameters, has a better fit than the Pareto, but not as good as the 4-parameter distribution. Even though the four parameters has weakened the simplicity, the bias and RMSE is quite low and the model provides a good fit to the Gamma distributed sample. It is also worth noting that the 4-parameter is slightly overestimating the reserve. In general insurance it is a good idea to rather risk overestimation than to risk underestimation. This welcomes a conservative risk strategy.

Table 4.3: True distribution: Weibull ( $n = 5000$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	2.101	0.017	36.705	2.721	1.053
	$\widehat{\text{RMSE}}$	2.126	0.295	36.768	2.863	2.674
$\epsilon = 0.01$	$\hat{b}$	3.219	0.018	58.639	3.403	1.272
	$\widehat{\text{RMSE}}$	3.241	0.323	58.788	3.569	3.009

In Table 4.3, the biases and root mean squared errors are given when Weibull is the true distribution. The results are very much similar to the results in Table 4.2, where Gamma was the true distribution. The Gamma distribution fits the data well, but not as well as the Weibull distribution fitted the Gamma data in Table 4.2. This could be explained by the slightly smaller tails in the Weibull distribution than in the Gamma distribution (see Figure 3.1). Using the same logic, we can see that the Pareto distribution provides an even worse fit than earlier. The extended Pareto distribution still has a good fit where the extra parameter does not seem to make a big difference for the RMSE. The 4-parameter distribution provides the best fit of all, except the true distribution, with the lowest bias. Again, the 4-parameter model is slightly overestimating the reserve.

Table 4.4: True distribution: Pareto ( $n = 5\,000$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-4.386	-4.333	0.025	0.027	0.058
	$\widehat{\text{RMSE}}$	4.638	4.571	1.941	1.949	2.062
$\epsilon = 0.01$	$\hat{b}$	-9.932	-9.416	0.071	0.095	0.243
	$\widehat{\text{RMSE}}$	10.082	9.570	3.060	3.148	3.768

When Pareto is the true distribution, as in Table 4.4, we can see that the more light-tailed distributions Gamma and Weibull do not fit the data very well. They are both underestimating the reserve which can have a major impact on the company's financial situation. At the same time, this is as expected because of the different tails. The extended Pareto and the 4-parameter distributions, however, fit the data exceptionally well. Hence, the 3- and 4-parameter model is a very good choice when we have a large heavy-tailed sample. Also, their RMSE is not that large, justifying the use of the extra parameters.

Table 4.5: True distribution: extended Pareto ( $n = 5\,000$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-3.325	-2.709	-1.720	0.090	0.062
	$\widehat{\text{RMSE}}$	3.548	2.984	2.122	1.469	1.583
$\epsilon = 0.01$	$\hat{b}$	-7.282	-6.407	-4.489	0.162	0.140
	$\widehat{\text{RMSE}}$	7.417	6.569	4.732	2.188	2.641

In Table 4.5 the biases and root mean squared errors are given when extended Pareto is the true distribution. The Gamma and Weibull distributions do not fit these data very well and are both underestimating the reserve. The Weibull distribution fits the data to some degree better than the Gamma distribution. The Pareto distribution is also underestimating the reserve, but is preferable to the Gamma and Weibull distributions. Moreover, the 4-parameter distribution provides a really good fit of the true distribution, for both the  $q_{0.95}$  and  $q_{0.99}$  reserves.

Table 4.6: True distribution: 4 parameter ( $n = 5000$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-5.930	-5.571	-4.203	1.551	-0.043
	$\widehat{\text{RMSE}}$	6.201	5.814	4.502	2.807	2.422
$\epsilon = 0.01$	$\hat{b}$	-14.622	-13.881	-10.710	4.748	0.091
	$\widehat{\text{RMSE}}$	14.769	14.019	10.919	6.586	5.071

When the 4-parameter is the true distribution we get the quantities as in Table 4.6. Gamma, Weibull, Pareto and extended Pareto are special cases of the 4 parameter distribution. These special cases have less parameters, and consequently less flexibility. We can see that they do not fit the true distribution as well as the 4-parameter model fit the true distribution. The extended Pareto provides a reasonably good fit with a slightly overestimating of the reserve. The two-parameter models, however, are all underestimating the reserve and do not fit the true distributed data very well. This underestimation can make it tough to cover the liabilities for losses.

#### 4.1.2 Sample Size: 500

For a sample size  $n = 500$ , we still have a substantial amount of observations, but the variation can be slightly larger when calculating the maximum likelihood estimated parameters. This can further lead to more varying reserves, and greater bias and RMSE.

Table 4.7: True distribution: Gamma ( $n = 500$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	0.077	0.116	17.666	5.289	0.970
	$\widehat{\text{RMSE}}$	1.542	1.572	17.911	8.867	3.265
$\epsilon = 0.01$	$\hat{b}$	0.146	0.104	25.610	8.506	1.261
	$\widehat{\text{RMSE}}$	1.694	1.729	25.961	14.495	3.770

When Gamma is the true distribution we get the biases and root mean squared errors as presented in Table 4.7. The results are very much comparable with the results in Table 4.2, although the bias and RMSE are a bit larger for Gamma, Weibull, Pareto and extended Pareto. This is as

expected since we have a smaller sample size. For the 4-parameter model, however, the bias and RMSE are smaller than for  $n = 5000$ . This could simply be a coincidence or an effect of the true distribution being more uncertain, which the added flexibility of the 4-parameter distribution handles well.

Table 4.8: True distribution: Weibull ( $n = 500$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	2.254	-0.048	41.390	1.864	0.551
	$\widehat{RMSE}$	2.433	0.877	41.559	2.396	2.274
$\epsilon = 0.01$	$\hat{b}$	3.493	-0.052	68.521	2.357	0.677
	$\widehat{RMSE}$	3.639	0.950	68.915	2.972	2.526

In Table 4.8, Weibull is the true distribution. The biases and root mean squared errors are approximately the same as in Table 4.3 when  $n = 5000$ . And again, the extended Pareto and 4-parameter model provide an even better fit when the sample is  $n = 500$ . We especially notice how well the 4-parameter model fits the Weibull distributed samples.

Table 4.9: True distribution: Pareto ( $n = 500$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-4.680	-4.649	0.008	0.013	0.383
	$\widehat{RMSE}$	6.746	6.607	6.524	6.543	7.280
$\epsilon = 0.01$	$\hat{b}$	-10.252	-9.778	0.647	0.753	2.557
	$\widehat{RMSE}$	11.679	11.229	10.948	11.281	15.412

When Pareto is the true distribution we get the biases and root mean squared errors presented in Table 4.9. The Gamma and Weibull distributions underestimate the reserve and do not produce a good fit of the data. On the other hand, the extended Pareto and 4-parameter distribution fit the data well, but not as well as for the  $n = 5000$  case. We know that the Pareto distribution is a very heavy-tailed distribution, and with an even lower sample size, the greatness of the tail is tough to determine. We can clearly see this from in the 4-parameter case. The bias and RMSE jumps from  $\hat{b}_{4P} = 0.383$  to  $\hat{b}_{4P} = 2.557$  and from  $\widehat{RMSE}_{4P} = 7.280$  to  $\widehat{RMSE}_{4P} = 15.412$ , respectively, when we consider the 95% percentile and



the 99% percentile. Hence, we could say that the 4-parameter model fits reasonably well up to some threshold.

Table 4.10: True distribution: extended Pareto ( $n = 500$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-3.447	-2.895	-1.369	0.038	0.244
	$\widehat{\text{RMSE}}$	5.126	4.782	3.854	4.770	5.170
$\epsilon = 0.01$	$\hat{b}$	-7.414	-6.620	-3.872	0.389	1.319
	$\widehat{\text{RMSE}}$	8.576	7.941	5.759	7.414	9.575

In Table 4.10 the biases and root mean squared errors are presented when extended Pareto is the true distribution. Now, the Gamma and Weibull distributions fit the data a little bit better than they did when the two parameter model, Pareto, was the true distribution in Table 4.9. The Pareto distribution gives a reasonably good fit for  $\epsilon = 0.05$  and  $\epsilon = 0.01$ . Moreover, the 4-parameter model has a very low bias, but because of the added uncertainty coming as a result of the smaller sample size, the RMSEs are not that good. However, it produces the best fit of all the models, except the true distribution.

Table 4.11: True distribution: 4 parameter ( $n = 500$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-6.031	-5.763	-4.148	1.918	0.704
	$\widehat{\text{RMSE}}$	7.864	7.458	6.523	7.742	8.190
$\epsilon = 0.01$	$\hat{b}$	-14.778	-14.166	-10.537	6.089	3.380
	$\widehat{\text{RMSE}}$	15.872	15.214	12.514	16.382	19.707

When the 4-parameter is the true distribution we get the quantities presented in Table 4.11. Other than the 4-parameter model, none of the distributions seems to provide a good fit when the sample comes from a 4-parameter distribution. All the two-parameter models are underestimating the reserve. It is the 4-parameter model which has the best fit, with the extended Pareto model not that far behind. It is interesting that the RMSEs are smaller for the 3-parameter model than for the 4-parameter model. Again, this could be a result of the lesser sample size, making it harder to, especially, determine the tail of the distribution.

### 4.1.3 Sample Size: 50

For a sample size of just  $n = 50$  it is very difficult to find the maximum likelihood estimates in the five models. The variability could be enormous and the shape parameters can be overestimated. Furthermore this will create large uncertainty in the reserve estimates, and corresponding large values of root mean squared errors since it is especially sensitive to large values.

Table 4.12: True distribution: Gamma ( $n = 50$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	0.628	0.153	25.390	8.348	0.943
	$\widehat{\text{RMSE}}$	4.977	5.090	26.848	15.975	5.513
$\epsilon = 0.01$	$\hat{b}$	0.951	0.111	39.080	15.160	1.325
	$\widehat{\text{RMSE}}$	5.452	5.593	41.139	32.230	6.382

In Table 4.12, the biases and root mean squared errors are presented for the five distributions when Gamma is the true distribution. Obviously, from  $\hat{b}_W = 0.176$  and  $\hat{b}_W = 0.147$  the Weibull model fits very well in the whole tail. At the same time we observe that the corresponding RMSEs are quite large. This probably means that we have more uncertainty than earlier when the sample size was larger. The Pareto model produces a really poor fit of the Gamma distributed sample. This means that the extreme right tail is exposed to overestimation as this model assumes more large claims than reasonable. The extended Pareto distribution has a bit better fit, but it is still not an adequate model to use. Finally, the 4-parameter distribution provides a good fit bearing in mind the small sample size.

Table 4.13: True distribution: Weibull ( $n = 50$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	2.693	-0.003	51.544	1.218	-0.270
	$\widehat{\text{RMSE}}$	3.953	2.787	52.443	3.682	6.214
$\epsilon = 0.01$	$\hat{b}$	4.138	-0.016	91.498	1.645	-0.248
	$\widehat{\text{RMSE}}$	5.222	3.016	93.463	5.493	6.688

In Table 4.13, the biases and root mean squared errors have been calculated with Weibull as the true distribution. The results are very much like

the results in Table 4.12. The biggest difference is that the Pareto model has an even worse fit than when the sample was Gamma distributed. At the same time, the extended Pareto model has a better fit. The Weibull distribution has a more concentrated shape than Gamma which makes the Pareto model a poor choice of model (see Figure 3.1). The 4-parameter distribution produces a really good fit based on the bias  $\hat{b}_{4P} = -0.270$  and  $\widehat{RMSE}_{4P} = -0.248$ . Because of the small sample size, the RMSEs tells us that we must use this model with caution. Still, the 4-parameter is a good choice, at least for the more light-tailed distributions.

Table 4.14: True distribution: Pareto ( $n = 50$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-4.183	-4.195	5.532	19.388	7.580
	$\widehat{RMSE}$	14.934	14.621	25.335	98.322	31.564
$\epsilon = 0.01$	$\hat{b}$	-9.759	-9.331	15.830	94.475	26.474
	$\widehat{RMSE}$	19.110	18.813	59.691	577.862	102.008

Table 4.15: True distribution: Pareto ( $n = 50$ ) without extreme reserve estimates

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-4.367	-4.346	4.672	7.702	5.665
	$\widehat{RMSE}$	14.709	14.415	23.642	33.448	26.519
$\epsilon = 0.01$	$\hat{b}$	-9.967	-9.511	13.364	27.118	19.034
	$\widehat{RMSE}$	18.924	18.621	53.025	102.328	76.839

When Pareto is the true distribution, we get the biases and root mean squared errors presented in Table 4.14. The moderate-tailed distributions, Gamma and Weibull, does not fit the data very well. This underestimation has been consistent for all sample sizes, although the RMSEs are a bit larger. For the more heavy-tailed distributions, the small sample size seems to create some complications. Even though the sample follows a Pareto distribution, the Pareto model struggles to fit the data. The extended Pareto model encounters the same problem. The issue is most noticeable far out in the right tail with a bias and RMSE of  $\hat{b}_{EP} = 94.475$  and  $\widehat{RMSE}_{EP} =$

577.862. There is also a large overestimation in the 4-parameter model, but much better than for the extended Pareto model.

There are a few extreme reserve estimates which pulls up the biases and RMSEs. When there are few data available, the `optim()` function is struggling to find the maximum likelihood estimates. This can make extreme reserve estimates. If we ignore the 20 – 30 largest reserve estimates we get the quantities as in Table 4.15. All models are still providing great uncertainty in cases of heavy tails and few sample data. One can hardly expect anything better.

Table 4.16: True distribution: extended Pareto ( $n = 50$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-2.860	-3.104	2.590	69.116	5.193
	$\widehat{\text{RMSE}}$	25.252	16.192	14.597	184.707	25.928
$\epsilon = 0.01$	$\hat{b}$	-6.737	-6.886	2.719	320.522	17.722
	$\widehat{\text{RMSE}}$	30.048	20.122	22.469	1004.811	77.565

Table 4.17: True distribution: extended Pareto ( $n = 50$ ) without extreme reserve estimates

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-2.834	-3.036	2.027	22.700	4.676
	$\widehat{\text{RMSE}}$	25.032	15.849	14.040	53.024	23.747
$\epsilon = 0.01$	$\hat{b}$	-6.681	-6.757	1.937	75.968	15.459
	$\widehat{\text{RMSE}}$	29.806	19.740	21.929	186.961	66.756

In Table 4.16 and Table 4.17, the biases and root mean squared errors are presented for extended Pareto as the true distributions. The results have many similarities to the results in Table 4.14 and Table 4.15. The Gamma and Weibull distributions are underestimating the reserve, while the Pareto and 4-parameter distributions are overestimating the reserve. The 4-parameter model produces a slightly better fit of the sample distribution than in the previous tables. The heavy tails and few available data makes it hard to fit a model to the true distribution.

Table 4.18: True distribution: 4-parameter ( $n = 50$ )

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-6.599	-6.742	-1.618	55.640	9.357
	$\widehat{\text{RMSE}}$	16.910	15.809	16.749	161.694	40.313
$\epsilon = 0.01$	$\hat{b}$	-15.392	-15.291	-4.763	266.878	39.305
	$\widehat{\text{RMSE}}$	23.575	22.614	28.930	937.720	149.955

Table 4.19: True distribution: 4-parameter ( $n = 50$ ) without extreme reserve estimates

		Gamma	Weibull	Pareto	extPareto	4-parameter
$\epsilon = 0.05$	$\hat{b}$	-7.045	-7.005	-2.518	23.337	5.935
	$\widehat{\text{RMSE}}$	15.166	14.509	13.985	54.746	27.935
$\epsilon = 0.01$	$\hat{b}$	-15.915	-15.615	-6.835	85.994	24.486
	$\widehat{\text{RMSE}}$	22.110	21.458	20.972	202.773	90.516

Finally, in Table 4.18 and Table 4.19 the biases and root mean squared errors are given when the 4-parameter distribution is the true distribution. Because of their moderate tails, the Gamma and Weibull distributions underestimate the reserve. The Pareto model has the lowest bias, but it is also underestimating the reserve. The extended Pareto model has a really poor fit. With a sample of only  $n = 50$ , the maximum likelihood method could overestimate the parameters, which can produce extreme values of the reserve. Even the 4-parameter model is struggling to fit the true distribution.

## 4.2 Summary

In this chapter, we have presented the biases and root mean squared errors for each of the five distributions as the true distribution and for three different sample sizes. The most interesting part of the results is how well the 3- and 4-parameter distributions fit to the two parameter special cases. We add parameters to the distributions to make the model more flexible. At the same time, this weakens the simplicity of the model. However, the 4-parameter model fits the data very well indeed, both for the light-tailed and heavy-tailed distributions.

The 4-parameter even provides a better fit than the three parameter extended Pareto distribution which is struggling to fit the more light-tailed distributions. When the sample size decreases, the uncertainty in the reserve estimates increases. The 4-parameter model has a reasonable good fit all over for sample size  $n = 5\,000$  and  $n = 500$ . When  $n = 500$  the extended Pareto model provides a slightly better fit for the heavy-tailed Pareto distribution. This is, however, not enough to exclude the 4-parameter model.

When the sample size is as low as  $n = 50$ , the tail becomes especially hard to capture. The parameters estimated by the maximum likelihood method can be overestimated, which creates reserve estimates that are far away from the true reserve. For the moderate-tailed distributions, the 4-parameter model is still very much applicable. When we have a more heavy-tailed sample, the 4-parameter model overestimates the reserve and hence does not fit the data that well. Be that as it may, perhaps our risk strategy is to be on the cautious side, which welcomes overestimation. It is a good idea to rather risk overestimation than to risk underestimation. At the same time, one should not overestimate too much as this binds capital.

# Chapter 5

## Conclusion

The total loss  $\mathcal{X}$  depends on two random variables, the claim frequency  $\mathcal{N}$  and the claim sizes  $Z_1, \dots, Z_{\mathcal{N}}$ . Claim sizes are typically modeled through simple two parameter distributions such as the Gamma, Weibull and Pareto distributions. An alternative is to use more flexible models which can be fitted to both light-tailed and heavy-tailed samples. An idea is to use the 4-parameter model, introduced in Chapter 2, with the previous mentioned distributions as special cases. In this way we can use only one model for different samples which can save us a lot of time, and more importantly, we can avoid choosing the wrong model that can have major consequences such as over- or underestimation.

We have used a simulation study to review how well the 3- and 4-parameter models estimate the reserve compared to the special cases Gamma, Weibull or Pareto distribution. The performance of the models was evaluated on the basis of the bias and root mean squared error of their reserve estimates.

We would like a model that is both simple, and at the same time provides a good fit of the true distribution. The moderate-tailed two-parameter models, Gamma and Weibull, tend to underestimate when fitted to a heavy-tailed sample. In the opposite case the heavy-tailed two parameter Pareto distribution overestimates the reserve when the sample follows a distribution with lighter tails. The added flexibility in the 3- and 4-parameter model handles the different shape and tails of the distributions much better. Even though the simplicity of the model is weakened when using the 4-parameter model instead of the extended Pareto model, it provides an even better fit in cases where the 3-parameter model is struggling to fit the more light-tailed distributions.

In general insurance it is crucial for the companies to set aside enough funds to cover future obligations. Therefore, when it comes to reserve es-

timation it is important not to underestimate the reserve. If the reserves prove to be inadequate in the future, we would have to increase reserves, which would reduce earnings and could have a material adverse effect on the insurance companies results of operations and financial condition. Therefore, it is a good idea to rather risk overestimation than to risk underestimation. At the same time, one should not overestimate too much as this may affect the affordability of insurance policies. Also, unnecessarily large reserve binds capital that may be better used to boost productivity and create employment [UNS, 2015]. From the results presented in tables in Chapter 4 the 4-parameter model is always slightly overestimating the reserve. Hence, the 4-parameter model is a conservative choice.

Combining the fact that the 4-parameter model has a good fit of the different types of samples and that it is a somewhat conservative choice, we can conclude that the ideas originally motivated for this thesis indeed has its justification. That is, that we can use the flexible 4-parameter model as a "standard model" to estimate the reserve when samples have different shape and tails. However, we must be cautious when handling small sample sizes.

When it comes to further work on the error analysis, it could be interesting to test another model for the claim frequency. The claim frequency  $\mathcal{N}$ , in this thesis, are assumed to follow a Poisson distribution. We could instead use a negative binomial distribution and investigate if the results differ. Also, we could check if a model with five or six parameters provide an even better fit than the 4-parameter model, and how well this model perform for small sample sizes.



# Bibliography

- [UNS, 2015] (2015). Are insurance companies holding too much capital? <https://www.businessthink.unsw.edu.au/Pages/Are-insurance-companies-holding-too-much-capital.aspx>. Accessed: 2017-05-24.
- [Arik and Umbleja, 2010] Arik, M. K. and Umbleja, M. (2010). Estimation of claim size distributions in estonian traffic insurance.
- [Bølviken, 2014] Bølviken, E. (2014). *Computation and modelling in insurance and finance*. Cambridge University Press.
- [Casella and Berger, 2002] Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.
- [Devore and Berk, 2011] Devore, J. L. and Berk, K. N. (2011). *Modern Mathematical Statistics with Applications*. Springer Science & Business Media.
- [Graham and Talay, 2013] Graham, C. and Talay, D. (2013). *Stochastic Simulation and Monte Carlo Methods: Mathematical Foundations of Stochastic Simulation*, volume 68. Springer Science & Business Media.
- [Gray and Pitts, 2012] Gray, R. J. and Pitts, S. M. (2012). *Risk modelling in general insurance: from principles to practice*. Cambridge University Press.
- [Hogg and Tanis, 2010] Hogg, R. V. and Tanis, E. A. (2010). *Probability and statistical inference*, volume 993.
- [Kaas et al., 2008] Kaas, R., Goovaerts, M., Dhaene, J., and Denuit, M. (2008). *Modern actuarial risk theory: using R*, volume 128. Springer Science & Business Media.
- [Panjer et al., 2008] Panjer, H. H., Klugman, S., and Willmot, G. (2008). *Loss models: From data to decisions*.

- [Ramsay, 2008] Ramsay, C. M. (2008). A dynamic model of a non-life insurance portfolio.
- [Ranka et al., 2010] Ranka, S., Banerjee, A., Biswas, K. K., Dua, S., Mishra, P., Moona, R., Poon, S.-H., and Wang, C.-L. (2010). *Contemporary Computing: Third International Conference, IC3 2010, Noida, India, August 9-11, 2010. Proceedings*, volume 94. Springer Science & Business Media.
- [Scholtes, 2001] Scholtes, S. (2001). The inverse transform method. <http://www3.eng.cam.ac.uk/~ss248/G12-M01/Week1/ITM.pdf>. Accessed: 2017-04-18.
- [Thomopoulos, 2012] Thomopoulos, N. T. (2012). *Essentials of Monte Carlo simulation: Statistical methods for building simulation models*. Springer Science & Business Media.

# Appendices



# Appendix A

## Mathematical Elaborations

### A.1 Probability

We start with some notation and definitions for basic quantities related to a random variable  $X$ . The *cumulative distribution function* (or just distribution function)  $F_X$  of  $X$ , is the probability that  $X$  will take a value less than or equal to  $x$ . Hence, it is given by

$$F_X(x) = \Pr(X \leq x), \quad x \in \mathbb{R}. \quad (\text{A.1})$$

The function  $F_X$  is non-decreasing and right-continuous [Gray and Pitts, 2012, p. 2]. Also,  $0 \leq F_X(x) \leq 1$  for all  $x \in \mathbb{R}$ ,  $\lim_{x \rightarrow \infty} F_X(x) = 1$  and  $\lim_{x \rightarrow -\infty} F_X(x) = 0$ .

A continuous random variable  $X$  has a *probability density function* (pdf)  $f_X$ , which is a non-negative function with  $\int_{-\infty}^{\infty} f_X(t) dt = 1$ , such that the distribution function of  $X$  is [Gray and Pitts, 2012, p. 2]

$$F_X(x) = \int_{-\infty}^x f_X(t) dt, \quad x \in \mathbb{R}, \quad (\text{A.2})$$

Hence,  $f_X(x) = \partial F_X(x) / \partial x$ . The probability that  $X$  takes on a value in the interval  $[a, b]$  is the area in this interval and under the density curve.

If we let  $X$  be a discrete random variable that takes values in  $\mathbb{N}$ , then  $\Pr(X = x), x \in \mathbb{R}$ , is the *probability mass function* of  $X$ . Since the pmf is a function that gives the probability that a discrete random variable  $X$  is exactly equal to some value  $x$ , we see that  $\Pr(X = x) = 0$  for  $x \notin \mathbb{N}$ . So for a discrete random variable concentrated on  $\mathbb{N}$ , the probability mass function is specified by  $\sum_{k=0}^{\infty} \Pr(N = k) = 1$ . Then, the distribution function of  $X$  is [Gray and Pitts, 2012, p. 3]

$$F_X(x) = \sum_{k \leq x} \Pr(X = k), \quad x \in \mathbb{R}. \quad (\text{A.3})$$

## A.2 Inversion

Assume that we can draw *uniform* random variables  $U$  for which all values between zero and one are equally likely to occur. By manipulating these uniforms we can generate other random variables  $X$  for a desired distribution with a cumulative distribution function  $\Pr(X \leq x) = F(x)$ .  $F(x)$  must be a strictly increasing and continuous distribution function. Then  $F(x) = u$  has the inverse solution  $x = F^{-1}(u)$ , and we define

$$X = F^{-1}(U) \quad \text{or} \quad X = F^{-1}(1 - U), \quad U \sim \text{uniform}. \quad (\text{A.4})$$

First, we notice that

$$\Pr(U \leq F(x)) = F(x)$$

for each  $x$  such that  $F(x) \in [0, 1]$ . Since  $F(X)$  is strictly increasing we have that

$$U \leq F(x) \Leftrightarrow F^{-1}(U) \leq x.$$

Hence the event  $U \leq F(x)$  occurs if and only if  $F^{-1}(U) \leq x$  and we conclude that [Scholtes, 2001]

$$\Pr(F^{-1}(U) \leq x) = \Pr(U \leq F(x)) = F(x).$$

Thus, we have a simple general sampling technique. However, it requires that we are able to calculate  $F^{-1}(\cdot)$ .

The cumulative distribution function of the Pareto distribution is

$$F(x) = 1 - (1 + x/\beta)^{-\alpha}, \quad x > 0. \quad (\text{A.5})$$

By solving the equation  $F(x) = u$  for  $x$  we get the inverse of the cdf

$$F^{-1}(u) = \beta((1 - u)^{-1/\alpha} - 1), \quad (\text{A.6})$$

which is the simple Pareto sampler.

### A.3 Extended Pareto: A Representation

Sampling from the extended Pareto distribution can be simplified by representing the extended Pareto variable  $Z$  as

$$Z = \frac{\theta\beta G_\theta}{\alpha G_\alpha}.$$

This can be shown by transformation. If we let  $Z = X/Y$  where  $X$  and  $Y$  are independent and positive random variables with density functions  $g_\theta(x)$  and  $g_\alpha(y)$ , respectively. Then, we have

$$F(z) = \Pr(Z \leq z) = \Pr(X \leq zY) = \int_0^\infty \Pr(X \leq zy)g_\alpha(y)dy.$$

Differentiating with respect to  $z$ , we get the density function of  $Z$

$$\begin{aligned} f(z) = F'(z) &= \frac{\partial}{\partial z} \int_0^\infty \Pr(X \leq zy)g_\alpha(y)dy \\ &= \int_0^\infty \frac{\partial}{\partial z} \Pr(X \leq zy)g_\alpha(y)dy \\ &= \int_0^\infty yg_\theta(zy)g_\alpha(y)dy, \end{aligned} \quad (\text{A.7})$$

since

$$\frac{\partial}{\partial z} \Pr(X \leq zy) = \frac{\partial}{\partial z} F_X(zy) = yF'_X(zy) = yg_\theta(zy).$$

Let  $X = \theta G_\theta$  and  $Y = \alpha G_\alpha$ , where  $G_\theta$  and  $G_\alpha$  are standard Gamma distributions with mean one and shape  $\theta > 0$  and  $\alpha > 0$ , respectively. Then, with  $g_\theta(x) = x^{\theta-1}e^{-x}/\Gamma(\theta)$  and  $g_\alpha(y) = y^{\alpha-1}e^{-y}/\Gamma(\alpha)$ ,

$$f(z) = \frac{z^{\theta-1}}{\Gamma(\theta)\Gamma(\alpha)} \int_0^\infty y^{\alpha+\theta-1}e^{-y(1+z)}dy = \frac{z^{\theta-1}}{\Gamma(\theta)\Gamma(\alpha)} \frac{1}{(1+z)^{\alpha+\theta}} \int_0^\infty w^{\alpha+\theta-1}e^{-w}dw$$

after substituting  $w = y(1+z)$  in the integral. Utilizing that  $\Gamma(s) = \int_0^\infty t^{s-1}e^{-t}dt$  we obtain

$$f(z) = \frac{\Gamma(\alpha + \theta)}{\Gamma(\theta)\Gamma(\alpha)} \frac{z^{\theta-1}}{(1+z)^{\alpha+\theta}}, \quad (\text{A.8})$$

which is the extended Pareto density (2.36) when  $\beta = 1$ .





# Appendix B

## R codes

### B.1 R code - Chapter 2

```
1
2 # ===== Poisson Distribution =====
3 # The probability mass functions are plotted
4
5 n=seq(0,40,1)
6 lambda=c(2,6,12)
7
8 plot(0,0,xlim=c(0,30),ylim=c(0,0.3),main="Poisson PMF","p",xlab="",
9      ylab="",xaxs="i",yaxs="i",cex.axis=1.5,cex.main=2)
10 legend("topright",legend=c("lambda=2","lambda=6","lambda=12"),pch=c(
11      0,1,2),lty=NA,lwd=1.5,box.lty=0,cex=1.5)
12 lines(n,dpois(n,lambda[1]),type="p",lwd=1.5,pch=0)
13 lines(n,dpois(n,lambda[2]),type="p",lwd=1.5,pch=1)
14 lines(n,dpois(n,lambda[3]),type="p",lwd=1.5,pch=2)
15 abline(h=0)
16
17 # ===== Gamma Distribution =====
18 density_gamma=function(m,alpha,xi){
19 # m simulations from the Gamma model with shape alpha and mean xi are
20 # generated and plotted.
21
22 plot(0,0,xlim=c(0,15),ylim=c(0,0.35),main="Gamma PDF","l",xlab="",
23      ylab="",xaxs="i",yaxs="i",cex.axis=1.5,cex.main=2)
24 legend("topright",legend=c("alpha=2, xi=3","alpha=4, xi=3","alpha
25      =4, xi=5"),lty=c(1,2,3,4),lwd=1.5,box.lty=0,cex=1.5)
26
27 for (i in 1:length(alpha)){
28   G=rgamma(m,alpha[i],alpha[i])
29   Z=xi[i]*G
30   d=density(Z,from=0,adjust=2)
31   lines(d$x,d$y,lty=i,type="l",lwd=1.5)
32 }
33 }
34
35 density_gamma(m=1000000,alpha=c(2,4,4),xi=c(3,3,5))
```

```

33
34
35 # ===== Weibull Distribution =====
36 density_weibull=function(m, alpha, beta){
37
38 # The Weibull density parameters shape alpha and scale beta are
39   estimated from m simulations and plotted.
40
41   plot(0,0, xlim=c(0,5), ylim=c(0,1.6), main="Weibull PDF", "1", xlab="",
42     ylab="", xaxs="i", yaxs="i", cex.axis=1.5, cex.main=2)
43   legend("topright", legend=c("alpha=2, beta=0.8", "alpha=3, beta=0.8",
44     "alpha=3, beta=1.5"), lty=c(1,2,3,4), lwd=1.5, box.lty=0, cex=1.5)
45
46   for (i in 1:length(alpha)){
47     Z=rweibull(m, alpha[i], beta[i])
48     d=density(Z, from=0, adjust=2)
49     lines(d$x, d$y, lty=i, type="l", lwd=1.5)
50   }
51 }
52
53 density_weibull(m=1000000, alpha=c(2,3,3), beta=c(0.8,0.8,1.5))
54
55 # ===== Pareto Distribution =====
56 density_pareto=function(m, alpha, beta){
57
58 # The Pareto density parameters shape alpha and scale beta are
59   estimated from m simulations and plotted.
60
61   plot(0,0, xlim=c(0,1.7), ylim=c(0,4.2), main="Pareto PDF", "1", xlab="",
62     ylab="", xaxs="i", yaxs="i", cex.axis=1.5, cex.main=2)
63   legend("topright", legend=c("alpha=10, beta=3", "alpha=15, beta=3",
64     "alpha=15, beta=6"), lty=c(1,2,3), lwd=1.5, box.lty=0, cex=1.5)
65
66   for (i in 1:length(alpha)){
67     U=runif(m,0,1)
68     Z=beta[i]*(U^(-1/alpha[i])-1)
69     d=density(Z, from=0, adjust=2)
70     lines(d$x, d$y, lty=i, type="l", lwd=1.5)
71   }
72 }
73
74 density_pareto(m=1000000, alpha=c(10,15,15), beta=c(3,3,6))
75
76 # ===== Extended Pareto Distribution =====
77 density_extPareto=function(m, alpha, theta, beta){
78
79 # m simulations of the Extended Pareto model with shape parameters
80   alpha and theta, and scale parameter beta are generated and
81   plotted.
82
83   plot(0,0, xlim=c(0,5), ylim=c(0,2.1), main="Extended Pareto PDF", "1",
84     xlab="", ylab="", xaxs="i", yaxs="i", cex.axis=1.5, cex.main=2)
85   legend("topright", legend=c("alpha=10, theta=3, beta=4", "alpha=30,
86     theta=3, beta=4", "alpha=30, theta=5, beta=4", "alpha=30, theta
87     =5, beta=7"), lty=c(1,2,3,4), lwd=1.5, box.lty=0, cex=1.5)
88
89   for (i in 1:length(alpha)){

```

```

81     X=theta [ i ] / alpha [ i ] *rgamma(m, theta [ i ] , theta [ i ] ) /rgamma(m, alpha [ i
      ], alpha [ i ] )
82     Z=beta [ i ] *X
83     d=density (Z, from=0, adjust=2)
84     lines (d$x, d$y, lty=i, type="l", lwd=1.5)
85   }
86 }
87
88 density_extPareto(m=1000000, alpha=c(10,30,30,30), theta=c(3,3,5,5),
      beta=c(4,4,4,7))
89
90
91 # ===== 4-parameter Distribution =====
92 density_4par=function(m, alpha, theta, beta, eta){
93
94 # m simulations from the 4-parameter model where the parameters alpha
      , theta, beta and eta are generated and plotted.
95
96 plot(0,0, xlim=c(0,25), ylim=c(0,0.3), main="4-parameter PDF", "l", xlab
      ="", ylab="", xaxs="i", yaxs="i", cex.axis=1.5, cex.main=2)
97 legend("topright", legend=c("alpha=10, theta=3, beta=4, eta=0.8", "
      alpha=30, theta=3, beta=4, eta=0.8", "alpha=30, theta=5, beta=4,
      eta=0.8", "alpha=30, theta=5, beta=7, eta=0.8", "alpha=30, theta
      =5, beta=7, eta=1.2"), lty=c(1,2,3,4,5), lwd=1.5, box.lty=0, cex
      =1.5)
98
99 for (i in 1:length(alpha)){
100   X=rgamma(m, theta [ i ] , theta [ i ] ) /rgamma(m, alpha [ i ] , alpha [ i ] )
101   Z=beta [ i ] *X^eta [ i ]
102   d=density (Z, from=0, adjust=2)
103   lines (d$x, d$y, lty=i, type="l", lwd=1.5)
104 }
105 }
106
107 density_4par(m=1000000, alpha=c(10,30,30,30,30), theta=c(3,3,5,5,5),
      beta=c(4,4,4,7,7), eta=c(0.8,0.8,0.8,0.8,1.2))

```

R\_code/R\_code\_chapter2.R

## B.2 R code - Chapter 3

```

1
2 # =====
3 # Finding suitable parameter values
4 # =====
5
6 # We plot the pdf for Gamma, Weibull, Pareto, extended Pareto and 4-
      parameter, and use their characteristics to find suitable
      parameters:
7
8 m=5000
9 plot(0,0, xlim=c(0,6), ylim=c(0,1.3), main="All pdf's", "l", xlab="", ylab=
      "", xaxs="i", yaxs="i", cex.axis=1.5, cex.main=2)
10 legend("topright", legend=c("Gamma", "Weibull", "Pareto", "ext Pareto"
      , "4 Parameter"), lty=c(1,2,3,4,5), lwd=1.5, box.lty=0, cex=1.5)

```

```

11
12 # Gamma:
13 alpha_g=3.4
14 xi_g=1
15 Z_g=xi_g*rgamma(m, alpha_g, alpha_g)
16 d=density(Z_g, from=0)
17 lines(d$x, d$y, lty=1, type="l", lwd=1.5)
18
19 # Weibull:
20 alpha_w=3.4
21 beta_w=1.1
22 Z_w=rweibull(m, alpha_w, beta_w)
23 d=density(Z_w, from=0)
24 lines(d$x, d$y, lty=2, type="l", lwd=1.5)
25
26 # Pareto:
27 alpha_p=3.4
28 beta_p=2.4
29 U=runif(m, 0, 1)
30 Z_p=beta_p*(U^(-1/alpha_p)-1)
31 d=density(Z_p, from=0)
32 lines(d$x, d$y, lty=3, type="l", lwd=1.5)
33
34 # Extended Pareto:
35 alpha_ep=3.4
36 theta_ep=2.0
37 beta_ep=1.2
38 Z_ep=beta_ep*theta_ep/alpha_ep*rgamma(m, theta_ep, theta_ep)/rgamma(m,
    alpha_ep, alpha_ep)
39 d=density(Z_ep, from=0)
40 lines(d$x, d$y, lty=4, type="l", lwd=1.5)
41
42 # 4-parameter:
43 alpha_4p=3.4
44 theta_4p=2.9
45 beta_4p=0.6
46 eta_4p=1.2
47 Z_4p=beta_4p*(rgamma(m, theta_4p, theta_4p)/rgamma(m, alpha_4p, alpha_4p)
    )^eta_4p
48 d=density(Z_4p, from=0)
49 lines(d$x, d$y, lty=5, type="l", lwd=1.5)
50
51
52 # -----
53 # Estimating the parameters
54 # -----
55
56 # ===== Gamma Distribution =====
57 nlogL_gamma=function(alpha, z){
58
59 # Function "nlogL_gamma" returns the negative of the log-likelihood:
60
61 -alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z)
    ))
62 }
63
64 parEstimate_gamma=function(z){
65

```

```

66 # The parameters alpha and xi are estimated where alpha is minimized
    by the optimize() function. The estimated parameters are returned
    by the parEstimate_gamma() function:
67
68 xi_hat=mean(z)
69 o=optimize(nlogL_gamma,c(0.0001,max(z)),z=z)
70 alpha_hat=o[[1]]
71 results=matrix(c(alpha_hat,xi_hat),1,2)
72 return(results)
73 }
74
75
76 # ===== Weibull Distribution =====
77 nlogL_weibull=function(alpha,z){
78
79 # Function "nlogL_weibull" returns the negative of the log-likelihood
    :
80
81 beta_alpha=mean(z**alpha)**(1/alpha)
82 -sum(dweibull(z,alpha,beta_alpha,log=T))
83 }
84
85 parEstimate_weibull=function(z){
86
87 # The parameters alpha and beta are estimated where alpha is
    minimized by the optimize() function. The estimated parameters
    are returned by the parEstimate_weibull() function:
88
89 o=optimize(nlogL_weibull,c(0.0001,50),z=z)
90 alpha_hat=o[[1]]
91 beta_hat=mean(z**alpha_hat)**(1/alpha_hat)
92 results=matrix(c(alpha_hat,beta_hat),1,2)
93 return(results)
94 }
95
96
97 # ===== Pareto Distribution =====
98 nlogL_pareto=function(beta,z){
99
100 # Function "nlogL_pareto" returns the negative of the log-likelihood:
101
102 alpha_beta=1/mean(log(1+z/beta))
103 -log(alpha_beta/beta)+(1+1/alpha_beta)
104 }
105
106 parEstimate_pareto=function(z){
107
108 # The parameters alpha and beta are estimated where beta is minimized
    by the optimize() function. The estimated parameters are
    returned by the parEstimate_pareto() function:
109
110 o=optimize(nlogL_pareto,c(0.0001,max(z)),z=z)
111 beta_hat=o[[1]]
112 alpha_hat=1/mean(log(1+z/beta_hat))
113 results=matrix(c(alpha_hat,beta_hat),1,2)
114 return(results)
115 }
116
117

```

```

118 # ===== Extended Pareto Distribution =====
119 nlogL_extPareto=function(s,z){
120
121 # Function "nlogL_extPareto" returns the negative of the log-
    likelihood:
122
123   t=exp(s)
124   -lgamma(t[1]+t[2])+lgamma(t[1])+lgamma(t[2])+t[2]*log(t[3])-(t
    [2]-1)*mean(log(z))+(t[1]+t[2])*mean(log(1+z/t[3]))
125 }
126
127 findAlpha=function(alpha,zBar,s2,skewZ){
128
129   theta=zBar^2*(alpha-1)/(s2*(alpha-2)-zBar^2)
130   2*sqrt((alpha-2)/(theta*(alpha+theta-1)))*(alpha+2*theta-1)/(alpha
    -3)-skewZ
131 }
132
133 parEstimate_extPareto=function(z){
134
135   # First we use the method of moments to find starting values for
    the optim() function:
136
137   zBar=mean(z)
138   s2=var(z)
139   skewZ=mean((z-zBar)^3)/s2^1.5
140
141   tmp=try(uniroot(findAlpha,interval=c(3.0000001,500),extendInt="yes",
    ,zBar=zBar,s2=s2,skewZ=skewZ)$root,silent=TRUE)
142   if(is.numeric(tmp)){
143     alpha.start=tmp
144   } else {
145     alpha.start=3.00001
146   }
147   if(s2*(alpha.start-2)-zBar^2<=0){
148     alpha.start=2+(0.1+zBar^2)/s2
149   }
150   theta.start=zBar^2*(alpha.start-1)/(s2*(alpha.start-2)-zBar^2)
151   beta.start=zBar*(alpha.start-1)/theta.start
152
153   # The parameters alpha, theta and beta are estimated by the optim()
    function which minimizes the nlogL_extPareto() function. We
    use alpha.start, theta.start and beta.start as starting values.
    The estimated parameters are returned by the parEstimate_
    extPareto() function:
154
155   o=optim(log(c(alpha.start,theta.start,beta.start)),nlogL_extPareto,
    z=z)
156   alpha_hat=exp(o$par[1])
157   theta_hat=exp(o$par[2])
158   beta_hat=exp(o$par[3])
159   results=matrix(c(alpha_hat,theta_hat,beta_hat),1,3)
160   return(results)
161 }
162
163
164
165
166

```

```

167 # ===== 4-parameter Distribution =====
168 nlogL_fourPar=function(s,z){
169
170 # Function "nlogL_fourPar" returns the negative of the log-likelihood
171 :
172 t=exp(s)
173 -lgamma(t[1]+t[2])+lgamma(t[1])+lgamma(t[2])-t[2]*log(t[2])+t[2]*
174   log(t[1])+(t[2]/t[4])*log(t[3])+log(t[4])-((t[2]/t[4])-1)*mean(
175     log(z))+(t[1]+t[2])*mean(log(1+(t[2]/t[1])*(z/t[3])^(1/t[4])))
176 }
177 parEstimate_fourPar=function(z){
178   # First we use the method of moments to find starting values for
179   # the optim() function:
180   zBar=mean(z)
181   s2=var(z)
182   skewZ=mean((z-zBar)^3)/s2^1.5
183
184   tmp=try(uniroot(findAlpha,interval=c(3.0000001,100),extendInt="yes"
185     ,zBar=zBar,s2=s2,skewZ=skewZ)$root,silent=TRUE)
186   if(is.numeric(tmp)){
187     alpha.start=tmp
188   } else {
189     alpha.start=3.00001
190   }
191   if(s2*(alpha.start-2)-zBar^2<=0){
192     alpha.start=2+(0.1+zBar^2)/s2
193   }
194   theta.start=zBar^2*(alpha.start-1)/(s2*(alpha.start-2)-zBar^2)
195   beta.start=zBar*(alpha.start-1)/alpha.start
196   eta.start=1
197 # The parameters alpha, theta, beta and eta are estimated by the
198 # optim() function which minimizes the nlogL_fourPar() function.
199 # The estimated parameters are returned by the parEstimate_fourPar
200 # () function:
201 o=optim(log(c(alpha.start,theta.start,beta.start,eta.start)),nlogL_
202   fourPar,z=z)
203 alpha_hat=exp(o$par[1])
204 theta_hat=exp(o$par[2])
205 beta_hat=exp(o$par[3])
206 eta_hat=exp(o$par[4])
207 results=matrix(c(alpha_hat,theta_hat,beta_hat,eta_hat),1,4)
208 return(results)
209 }
210
211
212
213
214
215
216

```

```

217 # -----
218 # Estimating the reserve
219 # -----
220
221 # ===== Gamma Distribution =====
222 reserve_gamma=function(m,JuT,alpha,xi,epsilon){
223
224 # The reserve with solvency level epsilon is estimated for a Poisson/
    Gamma portfolio with JuT as the expected number of claims, and m
    is the number of simulations used. The parameters alpha and xi
    are the estimated parameters when a certain distribution is
    fitted. The estimated reserve q-epsilon is returned by the
    function reserve_gamma():
225
226 N=rpois(m,JuT)
227 Z=array(0,m)
228
229 for(i in 1:m){
230   Z[i]=sum(xi*rgamma(N[i],alpha,alpha))
231 }
232
233 q_epsilon=array(0,length(epsilon))
234 for(i in 1:length(epsilon)){
235   q_epsilon[i]=sort(Z)[(1-epsilon[i])*m]
236 }
237 return(q_epsilon)
238 }
239
240
241 # ===== Weibull Distribution =====
242 reserve_weibull=function(m,JuT,alpha,beta,epsilon){
243
244 # The reserve with solvency level epsilon is estimated for a Poisson/
    Weibull portfolio with JuT as the expected number of claims, and
    m is the number of simulations used. The parameters alpha and
    beta are the estimated parameters when a certain distribution is
    fitted. The estimated reserve q-epsilon is returned by the
    function reserve_weibull():
245
246 N=rpois(m,JuT)
247 Z=array(0,m)
248
249 for(i in 1:m){
250   Z[i]=sum(rweibull(N[i],alpha,beta))
251 }
252
253 q_epsilon=array(0,length(epsilon))
254 for(i in 1:length(epsilon)){
255   q_epsilon[i]=sort(Z)[(1-epsilon[i])*m]
256 }
257 return(q_epsilon)
258 }
259
260
261 # ===== Pareto Distribution =====
262 reserve_pareto=function(m,JuT,alpha,beta,epsilon){
263
264 # The reserve with solvency level epsilon is estimated for a Poisson/
    Pareto portfolio with JuT as the expected number of claims, and m

```



```

    is the number of simulations used. The parameters alpha and beta
    are the estimated parameters when a certain distribution is
    fitted. The estimated reserve q-epsilon is returned by the
    function reserve_pareto():
265
266 N=rpois(m,JuT)
267 Z=array(0,m)
268
269 for(i in 1:m){
270   Z[i]=sum(beta*(runif(N[i]))**(-1/alpha)-1)
271 }
272
273 q_epsilon=array(0,length(epsilon))
274 for(i in 1:length(epsilon)){
275   q_epsilon[i]=sort(Z)[(1-epsilon[i])*m]
276 }
277 return(q_epsilon)
278 }
279
280
281 # ===== Extended Pareto Distribution =====
282 reserve_extPareto=function(m,JuT,alpha,theta,beta,epsilon){
283
284 # The reserve with solvency level epsilon is estimated for a Poisson/
    extended Pareto portfolio with JuT as the expected number of
    claims, and m is the number of simulations used. The parameters
    alpha, theta and beta are the estimated parameters when a certain
    distribution is fitted. The estimated reserve q-epsilon is
    returned by the function reserve_extPareto():
285
286 N=rpois(m,JuT)
287 Z=array(0,m)
288
289 for(i in 1:m){
290   Z[i]=sum(beta*theta/alpha*rgamma(N[i],theta,theta)/rgamma(N[i],
    alpha,alpha))
291 }
292
293 q_epsilon=array(0,length(epsilon))
294 for(i in 1:length(epsilon)){
295   q_epsilon[i]=sort(Z)[(1-epsilon[i])*m]
296 }
297 return(q_epsilon)
298 }
299
300
301 # ===== 4-parameter Distribution =====
302 reserve_fourPar=function(m,JuT,alpha,theta,beta,eta,epsilon){
303
304 # The reserve with solvency level epsilon is estimated for a Poisson/
    4-parameter portfolio with JuT as the expected number of claims,
    and m is the number of simulations used. The parameters alpha,
    theta, beta and eta are the estimated parameters when a certain
    distribution is fitted. The estimated reserve q-epsilon is
    returned by the function reserve_fourPar():
305
306 N=rpois(m,JuT)
307 Z=array(0,m)
308

```

```

309   for (i in 1:m){
310     Z[i]=sum(beta*(rgamma(N[i],theta,theta)/rgamma(N[i],alpha,alpha))
311           ^eta)
312   }
313   q_epsilon=array(0,length(epsilon))
314   for (i in 1:length(epsilon)){
315     q_epsilon[i]=sort(Z)[(1-epsilon[i])*m]
316   }
317   return(q_epsilon)
318 }
319
320
321 # -----
322 # Simulation study
323 # -----
324
325 reserve_all=function(z){
326
327 # Function "reserve_all" returns the estimated reserve of all
328   distributions Gamma, Weibull, Pareto, extended Pareto and 4-
329   parameter, when z follows the true distribution:
330
331   gamma_estimate=parEstimate_gamma(z)
332   res_gamma=reserve_gamma(m=1000000,JuT=50,alpha=gamma_estimate[1],xi
333     =gamma_estimate[2],epsilon=c(0.05,0.01))
334
335   weibull_estimate=parEstimate_weibull(z)
336   res_weibull=reserve_weibull(m=1000000,JuT=50,alpha=weibull_estimate
337     [1],beta=weibull_estimate[2],epsilon=c(0.05,0.01))
338
339   pareto_estimate=parEstimate_pareto(z)
340   res_pareto=reserve_pareto(m=1000000,JuT=50,alpha=pareto_estimate
341     [1],beta=pareto_estimate[2],epsilon=c(0.05,0.01))
342
343   extPareto_estimate=parEstimate_extPareto(z)
344   res_extPareto=reserve_extPareto(m=1000000,JuT=50,alpha=extPareto_
345     estimate[1],theta=extPareto_estimate[2],beta=extPareto_
346     estimate[3],epsilon=c(0.05,0.01))
347
348   fourPar_estimate=parEstimate_fourPar(z)
349   res_fourPar=reserve_fourPar(m=1000000,JuT=50,alpha=fourPar_estimate
350     [1],theta=fourPar_estimate[2],beta=fourPar_estimate[3],eta=
351     fourPar_estimate[4],epsilon=c(0.05,0.01))
352
353   return(c(res_gamma[1],res_weibull[1],res_pareto[1],res_extPareto
354     [1],res_fourPar[1],res_gamma[2],res_weibull[2],res_pareto[2],
355     res_extPareto[2],res_fourPar[2]))
356 }
357
358 # ===== Gamma - True Distribution =====
359 N=500
360 n=5000 # n=500, n=50, number of historical data
361 alpha=3.4
362 xi=1
363
364 reserve_trueGamma=matrix(NA,nrow=N,ncol=10)
365

```

```

356 for (i in 1:N){
357   z=xi*rgamma(n, alpha , alpha)
358   reserve_trueGamma[i,]=reserve_all(z)
359 }
360
361 # Creates a file "Reserve_trueGamma_5000.txt" containing the
      estimated reserve when Gamma is the true distribution:
362 write.table(reserve_trueGamma, file="Reserve_trueGamma_5000.txt", row.
      names=FALSE, col.names=c("Gamma.05", "Weibull.05", "Pareto.05", "
      extPar.05", "4Par.05", "Gamma.01", "Weibull.01", "Pareto.01", "extPar
      .01", "4Par.01"))
363
364
365 # ===== Weibull - True Distribution =====
366 N=500
367 n=5000 # n=500, n=50, number of historical data
368 alpha=3.4
369 beta=1.1
370
371 reserve_trueWeibull=matrix(NA, nrow=N, ncol=10)
372
373 for (i in 1:N){
374   z=rweibull(n, alpha , beta)
375   reserve_trueWeibull[i,]=reserve_all(z)
376 }
377
378 # Creates a file "Reserve_trueWeibull_5000.txt" containing the
      estimated reserve when Weibull is the true distribution:
379 write.table(reserve_trueWeibull, file="Reserve_trueWeibull_5000.txt",
      row.names=FALSE, col.names=c("Gamma.05", "Weibull.05", "Pareto.05", "
      extPar.05", "4Par.05", "Gamma.01", "Weibull.01", "Pareto.01", "extPar
      .01", "4Par.01"))
380
381
382 # ===== Pareto - True Distribution =====
383 N=500
384 n=5000 # n=500, n=50, number of historical data
385 alpha=3.4
386 beta=2.4
387
388 reserve_truePareto=matrix(NA, nrow=N, ncol=10)
389
390 for (i in 1:N){
391   z=beta*(runif(n)**(-1/alpha)-1)
392   reserve_truePareto[i,]=reserve_all(z)
393 }
394
395 # Creates a file "Reserve_truePareto_5000.txt" containing the
      estimated reserve when Pareto is the true distribution:
396 write.table(reserve_truePareto, file="Reserve_truePareto_5000.txt", row
      .names=FALSE, col.names=c("Gamma.05", "Weibull.05", "Pareto.05", "
      extPar.05", "4Par.05", "Gamma.01", "Weibull.01", "Pareto.01", "extPar
      .01", "4Par.01"))
397
398
399 # ===== Extended Pareto - True Distribution =====
400 N=500
401 n=5000 # n=500, n=50, number of historical data
402 alpha=3.4

```

```

403 theta=2.0
404 beta=1.2
405
406 reserve_trueExtPareto=matrix(NA,nrow=N,ncol=10)
407
408 for (i in 1:N){
409   z=beta*theta/alpha*rgamma(n,theta,theta)/rgamma(n,alpha,alpha)
410   reserve_trueExtPareto[i,]=reserve_all(z)
411 }
412
413 # Creates a file "Reserve_trueExtPareto_5000.txt" containing the
      estimated reserve when extended Pareto is the true distribution:
414 write.table(reserve_trueExtPareto,file="Reserve_trueExtPareto_5000.
      txt",row.names=FALSE,col.names=c("Gamma.05","Weibull.05","Pareto
      .05","extPar.05","4Par.05","Gamma.01","Weibull.01","Pareto.01",
      extPar.01","4Par.01"))
415
416
417 # ===== 4 Parameter - True Distribution =====
418 N=500
419 n=5000 # n=500, n=50, number of historical data
420 alpha=3.4
421 theta=2.9
422 beta=0.6
423 eta=1.2
424
425 reserve_trueFourPar=matrix(NA,nrow=N,ncol=10)
426
427 for (i in 1:N){
428   z=beta*(rgamma(n,theta,theta)/rgamma(n,alpha,alpha))^eta
429   reserve_trueFourPar[i,]=reserve_all(z)
430 }
431
432 # Creates a file "Reserve_trueFourPar_5000.txt" containing the
      estimated reserve when 4-parameter is the true distribution:
433 write.table(reserve_trueFourPar,file="Reserve_trueFourPar_5000.txt",
      row.names=FALSE,col.names=c("Gamma.05","Weibull.05","Pareto.05",
      extPar.05","4Par.05","Gamma.01","Weibull.01","Pareto.01",
      extPar.01","4Par.01"))

```

R\_code/R\_code\_chapter3.R

### B.3 R code - Chapter 4

```

1
2 # =====
3 # Calculate the bias and root mean squared error
4 # =====
5
6 # Functions that takes the true reserve and the estimated reserves as
      input and calculate the biases:
7
8 bias.05=function(my_data,true_reserve){ # epsilon=0.05
9   b_hat=array(0,5)
10  for (j in 1:5){

```

```

11     b_hat[j]=1/500*sum(my_data[,j]-true_reserve)
12   }
13   return(b_hat)
14 }
15
16 bias.01=function(my_data,true_reserve){ # epsilon=0.01
17   b_hat=array(0,5)
18   for(j in 1:5){
19     b_hat[j]=1/500*sum(my_data[,5+j]-true_reserve)
20   }
21   return(b_hat)
22 }
23
24
25 # Functions that takes the true reserve and the estimated reserves as
    input and calculate the root mean squared errors:
26
27 RMSE.05=function(my_data,true_reserve){ # epsilon=0.05
28   RMSE_hat=array(0,5)
29   for(j in 1:5){
30     RMSE_hat[j]=sqrt(1/500*sum((my_data[,j]-true_reserve)^2))
31   }
32   return(RMSE_hat)
33 }
34
35 RMSE.01=function(my_data,true_reserve){ # epsilon=0.01
36   RMSE_hat=array(0,5)
37   for(j in 1:5){
38     RMSE_hat[j]=sqrt(1/500*sum((my_data[,5+j]-true_reserve)^2))
39   }
40   return(RMSE_hat)
41 }
42
43
44 # ===== Gamma Distribution =====
45 Reserve_trueGamma_5000=read.table("/Users/simenholtter/Documents/Skole
    /MastersThesis2017/Reserve/Reserve_trueGamma_5000.txt",header=T)
46
47 alpha=3.4
48 xi=1
49 true_reserve=reserve_gamma(m=10000000,JuT=50,alpha,xi,epsilon=c
    (0.05,0.01))
50 write.table(true_reserve,file="True_Reserve_Gamma.txt",row.names=
    FALSE,col.names=FALSE)
51 #true_reserve=read.table("/Users/simenholtter/Documents/Skole/
    MastersThesis2017/Reserve/True_Reserve_Gamma.txt",header=F)
52
53 true_reserve.05=true_reserve[1]
54 true_reserve.01=true_reserve[2]
55
56 # Calculate the bias and RMSE when n=5000:
57 round(bias.05(Reserve_trueGamma_5000,true_reserve.05),digits=3)
58 round(bias.01(Reserve_trueGamma_5000,true_reserve.01),digits=3)
59
60 round(RMSE.05(Reserve_trueGamma_5000,true_reserve.05),digits=3)
61 round(RMSE.01(Reserve_trueGamma_5000,true_reserve.01),digits=3)
62
63
64

```

```

65 # ===== Weibull Distribution =====
66 Reserve_trueWeibull_5000=read.table("/Users/simenholler/Documents/
      Skole/MastersThesis2017/Reserve/Reserve_trueWeibull_5000.txt",
      header=T)
67
68 alpha=3.4
69 beta=1.1
70 true_reserve=reserve_weibull(m=10000000,JuT=50,alpha,beta,epsilon=c
      (0.05,0.01))
71 write.table(true_reserve,file="True_Reserve_Weibull.txt",row.names=
      FALSE,col.names=FALSE)
72 #true_reserve=read.table("/Users/simenholler/Documents/Skole/
      MastersThesis2017/Reserve/True_Reserve_Weibull.txt",header=F)
73
74 true_reserve.05=true_reserve[1]
75 true_reserve.01=true_reserve[2]
76
77 # Calculate the bias and RMSE when n=5000:
78 round(bias.05(Reserve_trueWeibull_5000,true_reserve.05),digits=3)
79 round(bias.01(Reserve_trueWeibull_5000,true_reserve.01),digits=3)
80
81 round(RMSE.05(Reserve_trueWeibull_5000,true_reserve.05),digits=3)
82 round(RMSE.01(Reserve_trueWeibull_5000,true_reserve.01),digits=3)
83
84
85 # ===== Pareto Distribution =====
86 Reserve_truePareto_5000=read.table("/Users/simenholler/Documents/
      Skole/MastersThesis2017/Reserve/Reserve_truePareto_5000.txt",
      header=T)
87
88 alpha=3.4
89 beta=2.4
90 true_reserve=reserve_pareto(m=10000000,JuT=50,alpha,beta,epsilon=c
      (0.05,0.01))
91 write.table(true_reserve,file="True_Reserve_Pareto.txt",row.names=
      FALSE,col.names=FALSE)
92 #true_reserve=read.table("/Users/simenholler/Documents/Skole/
      MastersThesis2017/Reserve/True_Reserve_Pareto.txt",header=F)
93
94 true_reserve.05=true_reserve[1]
95 true_reserve.01=true_reserve[2]
96
97 # Calculate the bias and RMSE when n=5000:
98 round(bias.05(Reserve_truePareto_5000,true_reserve.05),digits=3)
99 round(bias.01(Reserve_truePareto_5000,true_reserve.01),digits=3)
100
101 round(RMSE.05(Reserve_truePareto_5000,true_reserve.05),digits=3)
102 round(RMSE.01(Reserve_truePareto_5000,true_reserve.01),digits=3)
103
104
105 # Alternatively, without extreme reserve estimates:
106 Reserve_truePareto_5000_2=read.table("/Users/simenholler/Documents/
      Skole/MastersThesis2017/Reserve/Reserve_truePareto_5000_2.txt",
      header=T)
107 true_reserve=read.table("/Users/simenholler/Documents/Skole/
      MastersThesis2017/Reserve/True_Reserve_Pareto.txt",header=F)
108
109 true_reserve.05=true_reserve[1,]
110 true_reserve.01=true_reserve[2,]

```

```

111 |
112 | # Calculate the bias and RMSE when n=5000:
113 | round(bias.05(Reserve_truePareto_5000_2,true_reserve.05),digits=3)
114 | round(bias.01(Reserve_truePareto_5000_2,true_reserve.01),digits=3)
115 |
116 | round(RMSE.05(Reserve_truePareto_5000_2,true_reserve.05),digits=3)
117 | round(RMSE.01(Reserve_truePareto_5000_2,true_reserve.01),digits=3)
118 |
119 |
120 | # ===== Extended Pareto Distribution =====
121 | Reserve_trueExtPareto_5000=read.table("/Users/simenholtzer/Documents/
      | Skole/MastersThesis2017/Reserve/Reserve_trueExtPareto_5000.txt",
      | header=T)
122 |
123 | alpha=3.4
124 | theta=2.0
125 | beta=1.2
126 | true_reserve=reserve_extPareto(m=10000000,JuT=50,alpha,theta,beta,
      | epsilon=c(0.05,0.01))
127 | write.table(true_reserve,file="True_Reserve_extPareto.txt",row.names=
      | FALSE,col.names=FALSE)
128 | #true_reserve=read.table("/Users/simenholtzer/Documents/Skole/
      | MastersThesis2017/Reserve/True_Reserve_extPareto.txt",header=F)
129 |
130 | true_reserve.05=true_reserve[1]
131 | true_reserve.01=true_reserve[2]
132 |
133 | # Calculate the bias and RMSE when n=5000:
134 | round(bias.05(Reserve_trueExtPareto_5000,true_reserve.05),digits=3)
135 | round(bias.01(Reserve_trueExtPareto_5000,true_reserve.01),digits=3)
136 |
137 | round(RMSE.05(Reserve_trueExtPareto_5000,true_reserve.05),digits=3)
138 | round(RMSE.01(Reserve_trueExtPareto_5000,true_reserve.01),digits=3)
139 |
140 |
141 | # Alternatively, without extreme reserve estimates:
142 | Reserve_trueExtPareto_5000_2=read.table("/Users/simenholtzer/Documents
      | /Skole/MastersThesis2017/Reserve/Reserve_trueExtPareto_5000_2.txt
      | ",header=T)
143 | true_reserve=read.table("/Users/simenholtzer/Documents/Skole/
      | MastersThesis2017/Reserve/True_Reserve_extPareto.txt",header=F)
144 |
145 | true_reserve.05=true_reserve[1,]
146 | true_reserve.01=true_reserve[2,]
147 |
148 | # Calculate the bias and RMSE when n=5000:
149 | round(bias.05(Reserve_trueExtPareto_5000_2,true_reserve.05),digits=3)
150 | round(bias.01(Reserve_trueExtPareto_5000_2,true_reserve.01),digits=3)
151 |
152 | round(RMSE.05(Reserve_trueExtPareto_5000_2,true_reserve.05),digits=3)
153 | round(RMSE.01(Reserve_trueExtPareto_5000_2,true_reserve.01),digits=3)
154 |
155 |
156 | # ===== 4-parameter Distribution =====
157 | Reserve_trueFourPar_5000=read.table("/Users/simenholtzer/Documents/
      | Skole/MastersThesis2017/Reserve/Reserve_trueFourPar_5000.txt",
      | header=T)
158 |
159 | alpha=3.4

```

```

160 theta=2.9
161 beta=0.6
162 eta=1.2
163 true_reserve=reserve_fourPar(m=1000000,JuT=50,alpha,theta,beta,eta,
  epsilon=c(0.05,0.01))
164 write.table(true_reserve,file="True_Reserve_fourPar.txt",row.names=
  FALSE,col.names=FALSE)
165 #true_reserve=read.table("/Users/simenholtter/Documents/Skole/
  MastersThesis2017/Reserve/True_Reserve_FourPar.txt",header=F)
166
167 true_reserve.05=true_reserve[1]
168 true_reserve.01=true_reserve[2]
169
170 # Calculate the bias and RMSE when n=5000:
171 round(bias.05(Reserve_trueFourPar_5000,true_reserve.05),digits=3)
172 round(bias.01(Reserve_trueFourPar_5000,true_reserve.01),digits=3)
173
174 round(RMSE.05(Reserve_trueFourPar_5000,true_reserve.05),digits=3)
175 round(RMSE.01(Reserve_trueFourPar_5000,true_reserve.01),digits=3)
176
177
178 # Alternatively, without extreme reserve estimates:
179 Reserve_trueFourPar_5000_2=read.table("/Users/simenholtter/Documents/
  Skole/MastersThesis2017/Reserve/Reserve_trueFourPar_5000_2.txt",
  header=T)
180 true_reserve=read.table("/Users/simenholtter/Documents/Skole/
  MastersThesis2017/Reserve/True_Reserve_FourPar.txt",header=F)
181
182 true_reserve.05=true_reserve[1,]
183 true_reserve.01=true_reserve[2,]
184
185 # Calculate the bias and RMSE when n=5000:
186 round(bias.05(Reserve_trueFourPar_5000_2,true_reserve.05),digits=3)
187 round(bias.01(Reserve_trueFourPar_5000_2,true_reserve.01),digits=3)
188
189 round(RMSE.05(Reserve_trueFourPar_5000_2,true_reserve.05),digits=3)
190 round(RMSE.01(Reserve_trueFourPar_5000_2,true_reserve.01),digits=3)

```

R\_code/R\_code\_chapter4.R