

A Study of the Loss Distribution of Natural Disasters in Norway

Comparing a Common Model with a Model Broken Down
into Catastrophe Types

Yue Xu

Master's Thesis , Spring 2016



Cover design by Martin Helsø

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.

Acknowledgement

First and foremost, I would like to thank Erik Bølviken for providing me with an interesting and challenging topic, and for giving me the data as well.

I would like to give a special and sincere thanks to my supervisor, Ingrid Hobæk Haff, for all the great assistance throughout this entire period. I appreciate so much for your patience and the time you have been taken to answer my questions, both during the meetings and by e-mails. I have always been grateful for your enthusiasm and advice.

I would also like to give a big thanks to all my dear friends for keeping me motivated. Your support has been very precious to me.

Last but not least, thanks to my parents for their love and encouragement.

Yue Xu

May 2016, Oslo

Contents

1	Introduction	1
1.1	Background	1
1.2	Layout	1
2	Theory and Models	3
2.1	Estimation of the Parameters	3
2.2	Catastrophe Frequency	4
2.2.1	Poisson Distribution	5
2.2.2	Negative Binomial Distribution	7
2.3	Amounts of Compensation	10
2.3.1	Parametric Distributions	10
2.3.1.1	Log-Normal Distribution	10
2.3.1.2	Gamma Distribution	11
2.3.1.3	Pareto Distribution	12
2.3.1.4	Extended Pareto Distribution	14
2.3.2	Non-Parametric Approach	15
2.3.3	Extreme Values	16
2.3.3.1	Determination of the Threshold	16
2.3.3.2	Mixed Model with Pareto	17
2.4	Model for Portfolio Loss	17
2.4.1	Uncertainty Evaluation Using Bootstrap	17
2.5	Reserve	18
2.5.1	Monte Carlo Simulation	19
2.5.2	Estimation of the Reserve	19
3	Data Analysis	22
3.1	Data	22
3.2	Common Model for All Catastrophe Types	23
3.2.1	Claim Frequency Modelling	23
3.2.2	Claim Size Modelling	24
3.3	Separate Models for Different Catastrophe Types	30
3.3.1	Catastrophe Type 1	30
3.3.1.1	Claim Frequency Modelling	30
3.3.1.2	Claim Size Modelling	31

3.3.2	Catastrophe Type 2 and Type 3	36
3.3.2.1	Claim Frequency Modelling	36
3.3.2.2	Claim Size Modelling	38
3.3.3	Catastrophe Type 4, Type 5 and Type 9	42
3.3.3.1	Claim Frequency Modelling	42
3.3.3.2	Claim Size Modelling	43
3.4	Summary	48
4	Total Loss and Reserve	51
4.1	Total Loss and Reserve Simulation	51
4.2	Reserve Comparison	51
4.3	Summary	53
5	Conclusion	54
	References	55
	Appendix A Mathematical Arguments	57
A.1	The Double Rules of Expectation and Variance	57
A.2	The Mean and Standard Deviation of the Negative Binomial Distribution	58
A.3	Inverse Transform Method	58
A.4	A Representation of the Extended Pareto	59
A.5	Kernel Density Estimation	60
A.5.1	Gaussian Kernel Method	60
A.6	Derivation of the Tail Distribution Function of Z	61
	Appendix B R-scripts	62

Chapter 1

Introduction

1.1 Background

Norway has been suffering from numerous natural disasters throughout its history. To compensate losses caused by natural disasters such as storm, flood, earthquake, avalanche or similar natural disasters, the Norwegian National Fund for Natural Damage Assistance (*Statens Naturskadefond*) was first established in 1961 by the Act on Natural Damage. However, the amounts of compensation had always been considerably limited. Therefore, an amendment of the Act of 1961 and the Act on Insurance Contracts was made on 8th June 1979. As a consequence, from 1st January 1980, the insurance against natural disasters was connected to the fire insurance. Thereafter, changes in the Natural Disaster Law and the Law of Insurance Agreements were made in 1980 [Eide, 2015]. The new Natural Disaster Insurance Law (*Lov om naturskadeforsikring*) was passed in 1989. According to the law, all buildings and accessories with insurance policies that cover fires are automatically insured against natural catastrophes. The arrangement is managed by Norwegian Natural Perils Pool (*Norsk Naturskadepool*), which is a compulsory coinsurance system with all Norwegian insurance companies as members. The Pool operates as a distribution pool, i.e. the insurance companies still have all contact with their own policy holders, the Pool only equalizes losses [NNP, 2016]. When natural disaster occurs, the member companies pay claims on their policies that exceed the deductible. The remaining loss is ceded into the Pool. The total Pool loss is then shared among the member companies according to their market share [de Seguros, 2008, pp.121–122]. The Norwegian Natural Perils Pool not only covers losses from natural catastrophes, but also creates a good platform for a systematic study with the different catastrophe types to prevent and reduce risks in the near future.

1.2 Layout

In chapter 2, a general introduction of the theory and models used in the thesis will be given. Possible distributions for the catastrophe frequency and the amounts of compensation will be introduced along with some important properties of the distributions. For the model of the amounts of compensation, both the parametric and the non-parametric models will be explained. Since extreme values might be involved, an introduction of the extreme value theory will be given as well. Thereafter, the model for the total loss will be given followed by an explanation of the bootstrap method for estimating the parameter

uncertainties. The reserve will be introduced in the last section of this chapter along with an explanation of the Monte Carlo simulation method. Possible sources of uncertainty will be discussed as well.

In chapter 3, the distributions for the model of the catastrophe frequency and the model of the amounts of compensation presented in chapter 2 will be applied to the data. The modelling will be divided into two parts. The first part is based on a common model by treating all catastrophes equally as single incidents whereas the second part consists of several individual models based on the catastrophe types. At the end of this chapter, a brief summary will be given discussing the best fitting distributions and some confidence intervals of the distribution parameters will be given as well.

In chapter 4, the model for the total loss will be constructed by using the best fitting distributions for the model of the catastrophe frequency and the amounts of compensation. Thereafter, estimates of the reserve will be calculated from the loss model. The reserve estimate based on the common model will be compared with the one based on a combination of the separate models.

In chapter 5, a conclusion will be given discussing which of the models should be chosen based on the results from the previous chapters.

Some useful mathematical arguments and the R-scripts are given in the appendix at the end.

Chapter 2

Theory and Models

In general insurance, the loss model usually consists of two components, the claim frequency and the claim size. In this thesis, they refer to the catastrophe occurrences and the amounts of compensation. The claim frequency is linked to the model for the claim number, i.e. the number of catastrophes. Denote the number of catastrophes as N or \mathcal{N} , where the calligraphic one is used at the portfolio level. The amounts of compensation are denoted as Z . By combining the models of \mathcal{N} and Z , we will obtain a model for the total loss \mathcal{X} . We assume that all Z_i 's follow the same probability distribution. Moreover, \mathcal{N} and Z_i 's are stochastically independent. In the given data, the catastrophe numbers and amounts of compensation are aggregated values for each of $J = 431$ municipalities (see Section 3.1). The portfolio in this setting therefore consists of the J municipalities.

2.1 Estimation of the Parameters

Before introducing the models, a quick explanation of how the parameters of the distributions are estimated will be given as follows.

One of the most usual ways of parameter estimation is the maximum likelihood method. Let X_1, \dots, X_n be some random variables. Based on some historical data x_1, \dots, x_n , the joint probability mass function (pmf) or probability density function (pdf) is given by:

$$f(x_1, \dots, x_n | \theta_1, \dots, \theta_m), \quad (2.1)$$

where $\theta_1, \dots, \theta_m$ are the unknown parameters that need to be estimated. The idea of the maximum likelihood is to keep the observed values x_1, \dots, x_n fixed while varying the parameters so that they maximize the joint pmf or pdf. Therefore equation (2.1) is now considered to be a function of $\theta_1, \dots, \theta_m$, and is called the *likelihood function*. In both theoretical studies and numerical computations the log-likelihood function is often used:

$$\mathcal{L}(\theta_1, \dots, \theta_m) = \log \{f(x_1, \dots, x_n | \theta_1, \dots, \theta_m)\}. \quad (2.2)$$

If x_1, \dots, x_n are independent from each other, (2.2) can also be written as:

$$\mathcal{L}(\theta_1, \dots, \theta_m) = \sum_{i=1}^n \log \{f(x_i | \theta_1, \dots, \theta_m)\}. \quad (2.3)$$

If there exists an analytical expression for the ML estimate, it can be found by taking the derivative of (2.2) to zero:

$$\left. \frac{\partial \mathcal{L}(\theta_1, \dots, \theta_m)}{\partial \theta_i} \right|_{\theta_i = \hat{\theta}_{i, \text{ML}}} = 0.$$

Otherwise, the log-likelihood function has to be maximized numerically using statistical softwares (one of the examples is the `optimize()` function in R).

When the sample size is large, the ML estimator $\hat{\theta}_{\text{ML}}$ is a consistent estimator of θ , which means that as the sample size increases, the estimates "converge" to the true values of the parameters being estimated. The ML estimator is unbiased, i.e. $E(\hat{\theta}_{\text{ML}}) = \theta$, and has the smallest variance among unbiased estimators under certain conditions, i.e. is asymptotically efficient (see [Lehmann, 1999]). Alternatively, other methods of parameter estimation such as the method of moment could also be used. However, the maximum likelihood method is usually preferable in general. The drawback of the method of moment is that there has to be as many equations as the unknown parameters, and the solutions of the equations are not necessarily unique and might be cumbersome to find. With the maximum likelihood method, elementary differentiation can often be used to derive the estimates, and it also gives better estimates in comparison to the method of moments [Bølviken, 2014, pp.238–240].

2.2 Catastrophe Frequency

The number of catastrophes will be modelled in two ways, either by a break-down into components model, i.e. a model based on the catastrophe types, or a common model including all the catastrophe incidents.

Denote the number of municipality as $j = 1, \dots, J$, the year as $k = 1, \dots, K$ and the catastrophe type as i . When modelling by the catastrophe types, the number of catastrophe occurrences for the type i is given by:

$$\mathcal{N}^i = \mathcal{N}_1^i + \dots + \mathcal{N}_K^i, \quad (2.4)$$

which is the aggregated occurrences in all J municipalities and during all K years. Each \mathcal{N}_k^i can be again written as:

$$\mathcal{N}_k^i = N_{1k}^i + \dots + N_{Jk}^i, \quad (2.5)$$

which is the aggregated occurrences in all the municipalities in the year k .

When treating all the catastrophes equally as a single type, the total number of occurrences is given by:

$$\mathcal{N}^{\text{Total}} = \mathcal{N}_1^{\text{Total}} + \dots + \mathcal{N}_K^{\text{Total}}, \quad (2.6)$$

where the number of occurrences of each year $\mathcal{N}_k^{\text{Total}}$ can be written as:

$$\mathcal{N}_k^{\text{Total}} = N_{1k}^{\text{Total}} + \dots + N_{Jk}^{\text{Total}}, \quad (2.7)$$

which is the sum of the occurrences from all the J municipalities.

The number of occurrences can usually be well described either by a Poisson or a negative binomial distribution. A more detailed explanation for these two distributions will be given below.

2.2.1 Poisson Distribution

The Poisson distribution is widely-used in probability theory and statistics to model the number of events occurring in a fixed time interval. The distribution is based on the following assumptions:

- *Independence.* The occurrences of catastrophes must be independent of each other.
- *Homogeneity.* The intensity of the catastrophes is assumed to be the same for all municipalities.
- The time period must be fixed. In our case, $T = 1$ year.

The probability mass function is given by:

$$\begin{aligned}\Pr(N = n) &= \frac{\lambda^n}{n!} e^{-\lambda} \\ &= \frac{(\mu T)^n}{n!} e^{-\mu T},\end{aligned}\tag{2.8}$$

with the parameter $\lambda = \mu T$. μ is the intensity which indicates the average occurrence of events in the given time interval. T is known as the time of exposure which indicates the time of individual's potential for accidents or other losses. In our case, it refers to the time of each municipality's potential for suffering from the natural disasters.

An illustration of the Poisson density with different λ values is given in figure 2.1. When λ is small, the density function tends to lie closer to 0 in the x-axis, meaning that 0 is more likely to get when the intensity is low. In our case, it indicates a higher probability of no catastrophe occurrence per year. As λ becomes larger, the center of the curve moves towards the right and hence more apart from 0, meaning that it is more unlikely to get 0 when there is a high intensity. In our case, it indicates a higher probability of suffering from the natural disasters per year. The shape of the density function changes as λ varies, the curve is more spread-out as λ becomes larger.

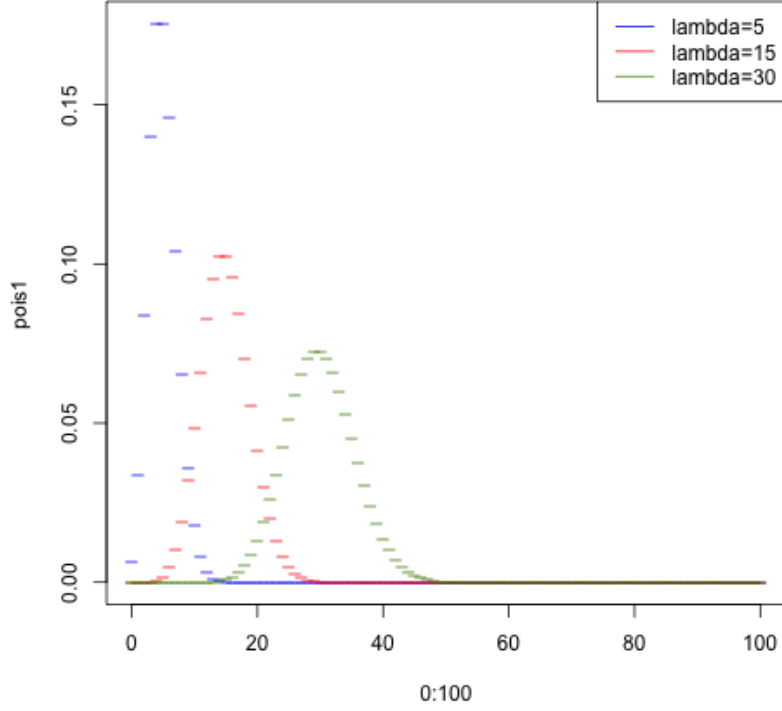


Figure 2.1: Some densities of the Poisson distribution as λ varies

The mean and the standard deviation are given by [Devore and Berk, 2007, p.145]:

$$E(N) = \lambda \quad \text{and} \quad \text{sd}(N) = \sqrt{\lambda}. \quad (2.9)$$

We assume that the number of occurrences in each municipality j of the year k , $N_{jk} \sim \text{Poisson}(\lambda)$ with parameter $\lambda = \mu T$ and $T = 1$ year. By the convolution property of the Poisson variables [Bølviken, 2014, p.283]:

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

where $N_1 \sim \text{Poisson}(\lambda_1), \dots, N_J \sim \text{Poisson}(\lambda_J)$. Followed by the relation (2.5) and (2.7), the intensity μ which describes how frequently the catastrophes occur, is assumed to be constant in all J municipalities over all K years. Hence the number of occurrences of the year k is given by:

$$\begin{aligned} \mathcal{N}_k^i &\sim \text{Poisson}(J\mu_i T) \quad \text{for catastrophe type } i, \\ \mathcal{N}_k^{\text{Total}} &\sim \text{Poisson}(J\mu_{\text{Total}} T) \quad \text{for all catastrophes.} \end{aligned}$$

Hence, the total number of claims during K years is given by:

$$\begin{aligned} \mathcal{N}^i &\sim \text{Poisson}(A\mu_i) \quad \text{for catastrophe type } i, \\ \mathcal{N}^{\text{Total}} &\sim \text{Poisson}(A\mu_{\text{Total}}) \quad \text{for all catastrophes,} \quad A = J \times K \times T. \end{aligned} \quad (2.10)$$

A is called the total risk exposure, which describes the municipalities' total potential for suffering from catastrophes. μ can be estimated for all N_{jk} 's simultaneously, and it can be determined from historical

data. Assume a total number of occurrences $n = n_1 + \dots + n_M$ of catastrophes from historical data, with a total risk exposure A , the Poisson probability is given by:

$$f(n|\mu) = \frac{(A\mu)^n}{n!} e^{-A\mu}. \quad (2.11)$$

The log-likelihood function is given by:

$$\mathcal{L}(\mu) = n \log(A\mu) - \log(n!) - A\mu.$$

Differentiating with respect to μ , we get the ML estimate of μ :

$$\hat{\mu} = \frac{n}{A} = \frac{n_1 + \dots + n_M}{A}. \quad (2.12)$$

The mean and the standard deviation of $\hat{\mu}$ are given by [Bølviken, 2014, p.284]:

$$E(\hat{\mu}) = \mu \quad \text{and} \quad \text{sd}(\hat{\mu}) = \sqrt{\frac{\mu}{A}}. \quad (2.13)$$

2.2.2 Negative Binomial Distribution

However, it is more natural to think that certain areas have a higher risk of suffering from catastrophes than other places in Norway. Now consider a conditional Poisson distribution such that:

$$[N_{jk}|\lambda_j] \sim \text{Poisson}(\lambda_j), \quad \text{where } \lambda_j = \mu_j T. \quad (2.14)$$

The frequency depends on the municipality, but is still constant from year to year. A common assumption for μ_j is that:

$$\mu_j = \xi G_j, \quad G_j \sim \text{Gamma}(\alpha), \quad (2.15)$$

where $\text{Gamma}(\alpha)$ is a standard Gamma distribution with mean 1 and shape α . The mean and the standard deviation of μ_j are given by [Bølviken, 2014, p.287]:

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

The conditional probability mass function is given by:

$$\Pr(N = n|\mu) = \frac{(\mu T)^n}{n!} e^{-\mu T}. \quad (2.17)$$

By the result from [Bølviken, 2014, p.286] and conditional probability [Ross, 2010, p.122]:

$$\Pr(N = n) = \int_0^\infty \Pr(N = n|\mu) g(\mu) d\mu,$$

where $g(\mu) = \frac{(\alpha/\xi)^\alpha}{\Gamma(\alpha)} \mu^{\alpha-1} e^{-\mu/\xi}$ which is the density function of Gamma. This leads to:

$$\begin{aligned} \Pr(N = n) &= \int_0^\infty \frac{(\mu T)^n}{n!} e^{-\mu T} \cdot \frac{(\alpha/\xi)^\alpha}{\Gamma(\alpha)} \mu^{\alpha-1} e^{-\mu/\xi} d\mu \\ &= \frac{T^n (\alpha/\xi)^\alpha}{n! \Gamma(\alpha)} \int_0^\infty \mu^{n+\alpha-1} e^{-\mu(T+\alpha/\xi)} d\mu. \end{aligned}$$

Substituting $z = \mu(T + \alpha/\xi)$ into the integral leads to:

$$\Pr(N = n) = \frac{T^n (\alpha/\xi)^\alpha}{n! \Gamma(\alpha) (T + \alpha/\xi)^{n+\alpha}} \int_0^\infty z^{n+\alpha-1} e^{-z} dz,$$

where the integral is $\Gamma(n + \alpha)$. Hence:

$$\begin{aligned}\Pr(N = n) &= \frac{\Gamma(n + \alpha)}{n!\Gamma(\alpha)} \cdot \frac{T^n(\alpha/\xi)^\alpha}{(T + \alpha/\xi)^{n+\alpha}} \\ &= \frac{\Gamma(n + \alpha)}{n!\Gamma(\alpha)} p^\alpha (1 - p)^n,\end{aligned}$$

where $p = \frac{\alpha}{\alpha + \xi T}$. With $\Gamma(n + 1) = n!$, the probability mass function can be now written as:

$$\Pr(N = n) = \frac{\Gamma(n + \alpha)}{\Gamma(n + 1)\Gamma(\alpha)} p^\alpha (1 - p)^n, \quad (2.18)$$

for $n = 0, 1, \dots$ [Bølviken, 2014, p.300]. Thus, we obtain a negative binomial distribution:

$$N_{jk} \sim \text{nb}in(\xi, \alpha), \quad (2.19)$$

with parameters ξ and α . Both ξ and α are independent of j , therefore we can estimate (ξ, α) for all N_{jk} 's simultaneously. The parameter α comes from the Gamma distribution, and it affects the shape of the distribution, see figure 2.2 below: the smaller the α is, the heavier tail the distribution has. In negative binomial distribution, α is also considered to be number of successes. Therefore as α becomes larger, the curve tends to move apart from 0, which indicates a larger number of catastrophe occurrences.

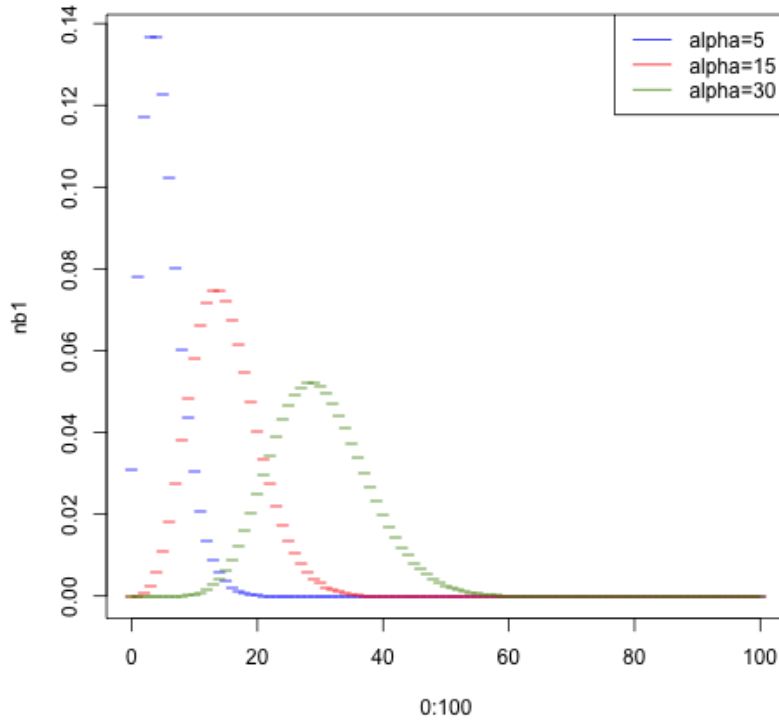


Figure 2.2: Some densities of the negative binomial distribution as α varies

The mean and the standard deviation are given by:

$$E(N) = \xi T \quad \text{and} \quad \text{sd}(N) = \sqrt{\xi T(1 + \xi T/\alpha)}, \quad (2.20)$$

which will be deduced in appendix A.2. As $\alpha \rightarrow \infty$, the variance converges to the mean, which means that the negative binomial distribution converges to the Poisson. Otherwise, the variance of the negative binomial distribution is always larger than the mean. Therefore the negative binomial is considered to be an extension of the Poisson that allows greater variance, and is more appropriate than the Poisson when overdispersion happens.

In our case, assume that N_{1k}, \dots, N_{Jk} are independent with common distribution $nbm(\xi, \alpha)$. Due to the convolution property of the negative binomial [Bølviken, 2014, p.301], the total number of occurrences of the year k is also negative binomial distributed:

$$\begin{aligned} \mathcal{N}_k^i &\sim nbm(J\xi_i, J\alpha_i) \quad \text{for catastrophe type } i, \\ \mathcal{N}_k^{\text{Total}} &\sim nbm(J\xi_{\text{Total}}, J\alpha_{\text{Total}}) \quad \text{for all catastrophes.} \end{aligned} \quad (2.21)$$

Hence, the total number of occurrences during K years is given by:

$$\begin{aligned} \mathcal{N}^i &\sim nbm(JK\xi_i, JK\alpha_i) \quad \text{for catastrophe type } i, \\ \mathcal{N}^{\text{Total}} &\sim nbm(JK\xi_{\text{Total}}, JK\alpha_{\text{Total}}) \quad \text{for all catastrophes.} \end{aligned} \quad (2.22)$$

Parameters ξ and α can be estimated using the maximum likelihood method, the log-likelihood function is given by:

$$\begin{aligned} \mathcal{L}(\xi, \alpha) &= \sum_{j=1}^M \log\{\Gamma(n_j + \alpha)\} - \sum_{j=1}^M \log\{\Gamma(n_j + 1)\} - n \log\{\Gamma(\alpha)\} \\ &\quad + \sum_{j=1}^M \alpha \log(p_j) + \sum_{j=1}^M n_j \log(1 - p_j) \\ &= \sum_{j=1}^M \log\{\Gamma(n_j + \alpha)\} - \sum_{j=1}^M \log\{\Gamma(n_j + 1)\} - n [\log\{\Gamma(\alpha)\} - \alpha \log(\alpha)] \\ &\quad + \sum_{j=1}^M n_j \log(\xi T_j) - \sum_{j=1}^M (n_j + \alpha) \log(\alpha + \xi T_j). \end{aligned} \quad (2.23)$$

As $T_j = 1$, $j = 1, \dots, M$, this simplifies to:

$$\begin{aligned} \mathcal{L}(\xi, \alpha) &= \sum_{j=1}^M \log\{\Gamma(n_j + \alpha)\} - \sum_{j=1}^M \log\{\Gamma(n_j + 1)\} - n [\log\{\Gamma(\alpha)\} - \alpha \log(\alpha)] \\ &\quad + \log(\xi) \sum_{j=1}^M n_j - n \alpha \log(\alpha + \xi) - \log(\alpha + \xi) \sum_{j=1}^M n_j. \end{aligned} \quad (2.24)$$

This means that:

$$\frac{\partial \mathcal{L}(\xi, \alpha)}{\partial \xi} = \frac{1}{\xi} \sum_{j=1}^M n_j - \frac{n\alpha}{\alpha + \xi} - \frac{1}{\alpha + \xi} \sum_{j=1}^M n_j = 0,$$

which gives:

$$\hat{\xi} = \frac{1}{n} \sum_{j=1}^M n_j = \bar{n}.$$

Inserting $\hat{\xi}$ into $\mathcal{L}(\xi, \alpha)$, we obtain a function of α only:

$$\begin{aligned} \mathcal{L}(\alpha) &= \sum_{j=1}^M \log\{\Gamma(n_j + \alpha)\} - \sum_{j=1}^M \log\{\Gamma(n_j + 1)\} - n [\log\{\Gamma(\alpha)\} - \alpha \log(\alpha)] \\ &\quad + \log(\bar{n}) \sum_{j=1}^M n_j - n \alpha \log(\alpha + \bar{n}) - \log(\alpha + \bar{n}) \sum_{j=1}^M n_j. \end{aligned} \quad (2.25)$$

With the help of statistical software such as R we can optimize $\mathcal{L}(\alpha)$ with respect to α numerically.

2.3 Amounts of Compensation

The amounts of compensation can be modelled either purely parametrically or semi-parametrically associated with the Pickand's theorem. Descriptions of both the parametric and the non-parametric distributions along with an explanation of Pickand's theorem are given as follows.

2.3.1 Parametric Distributions

All parametric models for the amounts of compensation are of the form:

$$Z = \beta Z_0, \quad \beta > 0, \quad (2.26)$$

where β is called the *parameter of scale* and Z_0 is a standardized random variable corresponding to $\beta = 1$. The benefit of expressing Z of this form is that if anything changes such as currency or inflation, the effect can always be passed to Z_0 by replacing β . In other words, the shape of the density function always remains what it was. The proportionality is also inherited by expectation and standard deviation [Bølviken, 2014, p.315]:

$$E(Z) = \beta E(Z_0) \quad \text{and} \quad \text{sd}(Z) = \beta \text{sd}(Z_0).$$

2.3.1.1 Log-Normal Distribution

Assume that $Z = \xi Z_0$, where $Z_0 = e^{-\sigma^2/2 + \sigma\varepsilon}$ for $\varepsilon \sim N(0, 1)$. Now let $Y = \log(Z)$, i.e.:

$$Y = \log(Z) = \log(\xi) - \frac{1}{2}\sigma^2 + \sigma\varepsilon. \quad (2.27)$$

$Y \sim N(\log(\xi) - \frac{1}{2}\sigma^2, \sigma^2)$ since it is a linear combination of ε . Hence, Z is said to be log-normal distributed. The probability density function of Z is given by [Devore and Berk, 2007, p.201]:

$$f(z) = \frac{1}{\sqrt{2\pi}\sigma z} e^{-\frac{(\log(z) - \mu)^2}{2\sigma^2}}, \quad \mu = \log(\xi) - \frac{1}{2}\sigma^2, \quad z > 0. \quad (2.28)$$

The mean and the standard deviation are given by [Bølviken, 2014, p.319]:

$$E(Z) = \xi \quad \text{and} \quad \text{sd}(Z) = \xi \sqrt{e^{\sigma^2} - 1}. \quad (2.29)$$

Assume there are historical data z_1, \dots, z_n , and let $y_i = \log(z_i)$. The estimates of the mean and the standard deviation of Y are given by:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad s_y = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2}. \quad (2.30)$$

By (2.27), we have the relations:

$$\bar{y} = \log(\hat{\xi}) - \frac{1}{2}\hat{\sigma}^2, \quad s_y = \hat{\sigma}. \quad (2.31)$$

Hence, the estimates of ξ and σ are given by:

$$\hat{\xi} = e^{s_y^2/2 + \bar{y}}, \quad \hat{\sigma} = s_y, \quad (2.32)$$

which are the ML estimates for ξ and σ , see [Ginos, 2009].

Figure 2.3 below illustrates the variation of the log-normal pdfs as σ and ξ varies. The shape of the pdf depends heavily on σ . As σ increases, the degree of skewness increases. For fixed σ , the pdf is more spread-out as ξ increases.

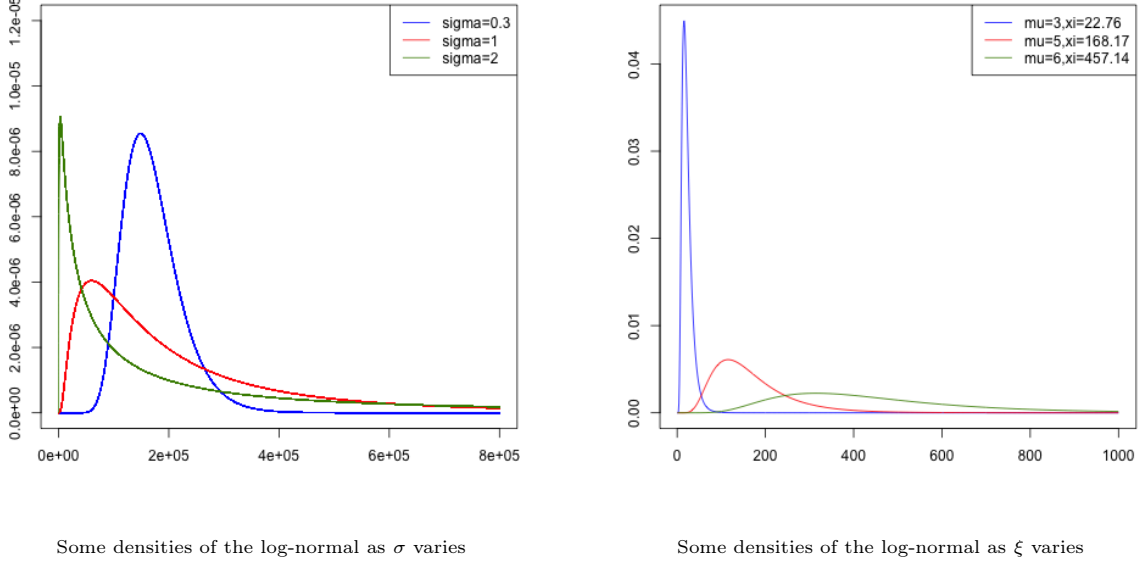


Figure 2.3: Comparison of the log-normal densities as the parameters varies

2.3.1.2 Gamma Distribution

We assume that $Z = \xi G$, where $G \sim \text{Gamma}(\alpha)$, i.e. a standard Gamma distribution with mean 1 and shape α . The probability density function of G is given by:

$$f(x; \alpha, \frac{1}{\alpha}) = \frac{1}{(1/\beta)^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x\alpha}, \quad x \geq 0.$$

By pdf inversion [Casella and Berger, 2002, p.51], the probability density function of Z can be written as:

$$f(z) = \frac{(\alpha/\xi)^\alpha}{\Gamma(\alpha)} z^{\alpha-1} e^{-\alpha z/\xi}, \quad z > 0. \quad (2.33)$$

The mean and the standard deviation are given by [Bølviken, 2014, p.41]:

$$E(Z) = \xi \quad \text{and} \quad \text{sd}(Z) = \frac{\xi}{\sqrt{\alpha}}. \quad (2.34)$$

Assume historical data z_1, \dots, z_n , the parameters can be estimated by the maximum likelihood method. The log-likelihood function is given by:

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

Differentiation with respect to ξ we obtain the ML estimate of ξ , which is given by:

$$\frac{\partial \mathcal{L}}{\partial \xi} = -\frac{n\alpha}{\xi} + \frac{\alpha}{\xi^2} \sum_{i=1}^n z_i = 0 \implies \hat{\xi} = \frac{\sum_{i=1}^n z_i}{n} = \bar{z}. \quad (2.35)$$

Now inserting the expression of $\hat{\xi}$ into the log-likelihood function, we will get a log-likelihood that only depends on α :

$$\mathcal{L}(\alpha) = n\alpha \log\left(\frac{\alpha}{\bar{z}}\right) - n \log\{\Gamma(\alpha)\} + (\alpha - 1) \sum_{i=1}^n \log(z_i) - \frac{\alpha}{\bar{z}} \sum_{i=1}^n z_i. \quad (2.36)$$

The optimization with respect to α must be done numerically in R.

Figure 2.4 illustrates the graphs of the Gamma pdf for several (α, ξ) pairs. With the same α , the graph is less right skewed as ξ becomes larger. With fixed ξ , the Gamma pdf is less spread-out as α increases. However, a change in ξ shows a sharper change on the pdf.

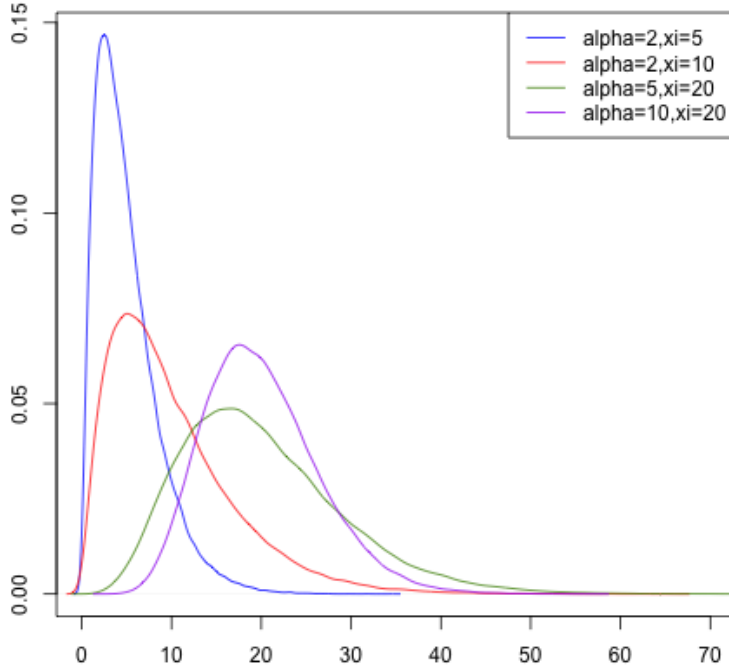


Figure 2.4: Some densities of the Gamma distribution as ξ and α vary

2.3.1.3 Pareto Distribution

The Pareto distribution is a very *heavy-tailed* distribution. Random variable with a heavy-tailed distribution takes very large values with a rather high probability. In our case, since extreme values can be involved, the Pareto distribution could be a conservative choice for the amounts of compensation model. Generally, the pdf of the Pareto distribution is defined as follows:

$$f(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}, \quad x \geq x_m, \quad (2.37)$$

where x_m is the minimum possible value that x can be. In our case, we assume the smallest possible value of x to be 1, hence (2.37) now becomes:

$$\frac{\alpha}{x^{\alpha+1}}, \quad x \geq 1. \quad (2.38)$$

Let $Z = \beta(x - 1)$, by inversion, the probability density function of Z is given by:

$$\begin{aligned} f_z(z) &= f_x\left(1 + \frac{z}{\beta}\right) \cdot \left|\frac{d}{dz}\left(1 + \frac{z}{\beta}\right)\right| \\ &= \frac{\alpha}{(1 + z/\beta)^{\alpha+1}} \cdot \frac{1}{\beta} \\ &= \frac{\alpha/\beta}{(1 + z/\beta)^{1+\alpha}}, \quad z > 0 \text{ and } \alpha, \beta > 0. \end{aligned} \tag{2.39}$$

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

$$E(Z) = \frac{\beta}{\alpha - 1} \quad \text{and} \quad \text{sd}(Z) = E(Z) \sqrt{\frac{\alpha}{\alpha - 2}}. \tag{2.40}$$

Estimates of the parameters can be obtained by the maximum likelihood method. The log-likelihood function is given by:

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

Differentiating with respect to α for given β we get:

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

which gives that:

$$\hat{\alpha}_\beta = \frac{n}{\sum_{i=1}^n \log\left(1 + \frac{z_i}{\beta}\right)}. \tag{2.41}$$

By inserting the expression of $\hat{\alpha}_\beta$ into the log-likelihood function, we get a likelihood function that only depends on β :

$$\begin{aligned} \mathcal{L}(\beta) &= n \log\left(\frac{\frac{n}{\sum_{i=1}^n \log(1 + z_i/\beta)}}{\beta}\right) - \left(1 + \frac{n}{\sum_{i=1}^n \log(1 + z_i/\beta)}\right) \sum_{i=1}^n \log\left(1 + \frac{z_i}{\beta}\right) \\ &= n \log\left(\frac{n}{\sum_{i=1}^n \log(1 + z_i/\beta)}\right) - n \log \beta - \sum_{i=1}^n \log\left(1 + \frac{z_i}{\beta}\right) - n. \end{aligned} \tag{2.42}$$

Again, the optimization with respect to β must be done numerically in R. The sampling method used for the Pareto is the inverse transform method, which will be justified in appendix A.3. Here the Pareto distribution is considered to be a special case of the extended Pareto which will be explained in detail in the following subsection.

Figure 2.5 illustrates the pdfs of the Pareto distribution for fixed β and α , respectively. As α increases with fixed β , the degree of skewness increases as well. The degree of skewness decreases as β increases with fixed α .

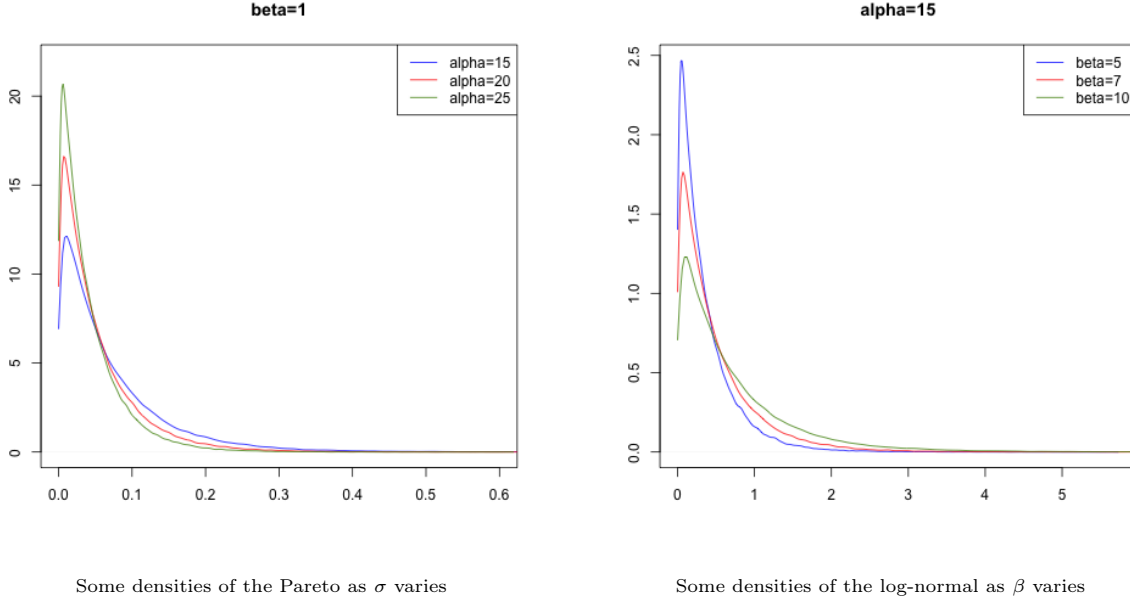


Figure 2.5: Comparison of the Pareto densities as the parameters varies

2.3.1.4 Extended Pareto Distribution

The extended Pareto distribution is a generalized version of the Pareto distribution presented in section 2.3.1.3 by setting $\theta = 1$. There is one more parameter, θ , in the extended Pareto. Therefore it is more flexible and hence may give a better fit to the data. The density function is given by:

$$f(z) = \frac{\Gamma(\alpha + \theta)}{\Gamma(\alpha)\Gamma(\theta)} \frac{1}{\beta} \frac{(z/\beta)^{\theta-1}}{(1 + z/\beta)^{\alpha+\theta}}, \quad z > 0, \alpha, \beta \text{ and } \theta > 0. \quad (2.43)$$

The distribution of the extended Pareto is complicated, but the variable Z with parameters (α, θ, β) can be expressed by:

$$Z = \frac{\theta\beta}{\alpha} \frac{G_1}{G_2}, \quad (2.44)$$

where $G_1 \sim \text{Gamma}(\theta)$, $G_2 \sim \text{Gamma}(\alpha)$ are two independent Gamma variables with mean 1. The representation will be justified in appendix A.4. By using (2.44), the sampling is simplified.

The mean and the standard deviation are given by [Bølviken, 2014, p.324]:

$$\text{E}(Z) = \frac{\theta\beta}{\alpha - 1} \quad \text{and} \quad \text{sd}(Z) = \text{E}(Z) \sqrt{\frac{\alpha + \theta - 1}{\theta(\alpha - 2)}}, \quad (2.45)$$

which are valid when $\alpha > 1$ and $\alpha > 2$, respectively.

The parameters can be estimated by the maximum likelihood method. The log-likelihood function is given by:

$$\begin{aligned} \mathcal{L}(\alpha, \theta, \beta) = & n[\log\{\Gamma(\alpha + \theta)\} - \log\{\Gamma(\alpha)\} - \log\{\Gamma(\theta)\} - \theta\log(\beta)] \\ & + (\theta - 1) \sum_{i=1}^n \log(z_i) - (\alpha + \theta) \sum_{i=1}^n \log\left(1 + \frac{z_i}{\beta}\right). \end{aligned} \quad (2.46)$$

The optimization of the log-likelihood of the extended Pareto is more complicated, and needs to be calculated numerically.

Figure 2.6 illustrates the variation of the extended Pareto pdfs as the three parameters vary. As α increases with fixed β and θ , the extended Pareto distribution is more right skewed. As θ increases with fixed α and β , the degree of skewness decreases. An increase in β yields a decrease in the degree of skewness. A change in β shows a sharper effect on the pdfs compared to the other two parameters.

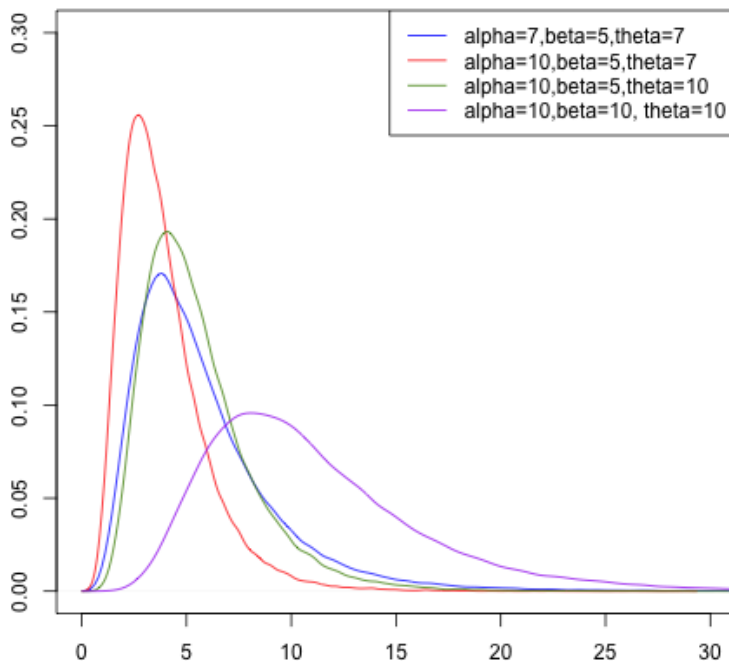


Figure 2.6: Comparison of the extended Pareto densities as the parameters vary

2.3.2 Non-Parametric Approach

When a parametric model does not describe the data well, a non-parametric model could be a good alternative for estimating future claims. Compared to the parametric model, the non-parametric one does not require the assumption about the distribution of the data, and is known as the empirical distribution. The cumulative distribution function of the empirical distribution is given by:

$$F_n(z) = \frac{1}{n} \sum_{i=1}^n I(Z_i \leq z), \quad (2.47)$$

where $I(\cdot)$ is the indicator function:

$$I(A) = \begin{cases} 1, & \text{if } A \text{ is true} \\ 0, & \text{otherwise.} \end{cases}$$

The probability density can be estimated using the **Gaussian kernel** method explained in appendix A.5.1. The mean and standard deviation of Z are given by:

$$E(Z) = \sum_{i=1}^n \frac{1}{n} z_i = \bar{z} \quad \text{and} \quad \text{sd}(Z) = \sqrt{\sum_{i=1}^n \frac{1}{n} (z_i - \bar{z})^2}. \quad (2.48)$$

With a large number of observations, the empirical distribution will usually be a good alternative.

2.3.3 Extreme Values

The amounts of compensation could be enormously large in our case. One of the distributions presented above can usually fit small or moderate claim sizes well, but the probability of claims with extreme large size could easily be underestimated by such models. The theory that has to be mentioned when it comes to extreme value estimation is the Pickands-Balkema-de Haan Theorem, also known as the extreme value theory. Let Z be continuous and unbounded, then the theorem states that there exists some threshold b such that its over-threshold model to the right of the limit, say Z_b , is guaranteed to be Pareto distributed as $b \rightarrow \infty$, regardless of what the original distribution was. Mathematically, $Z_b = Z - b | Z > b$, then the tail distribution function $\bar{F}_b(z) = \Pr(Z_b > z | Z > b)$ can be written as:

$$\bar{F}_b(z) = \frac{1 - F(b + z)}{1 - F(b)}, \quad \text{where } F(z) = \Pr(Z \leq z). \quad (2.49)$$

The distribution function will be derived in appendix A.6. Now let $Y_b = Z_b/\beta_b$, where β_b is a scale parameter depending on b . By the assumption that $Z > b$, the tail distribution is hence given by:

$$\Pr(Y_b > y) = \bar{F}_b(\beta_b y), \quad (2.50)$$

which will also be deduced in appendix A.6. Denote the tail distribution of the Pareto model as $\bar{P}(y/\alpha)$, where α is a parameter not depending on b and possibly infinite. Then the Pickand's theorem states that:

$$\bar{F}_b(\beta_b y) \rightarrow \bar{P}(y/\alpha) \quad \text{as } b \rightarrow \infty,$$

where

$$\bar{P}(y/\alpha) = \begin{cases} (1 + y)^{-\alpha}, & 0 < \alpha < \infty \\ e^{-y}, & \alpha = \infty. \end{cases} \quad (2.51)$$

Hence, $Z_b \sim \text{Pareto}(\alpha, \beta_b)$ as $b \rightarrow \infty$ with finite α or is exponentially distributed when α is infinite. The shape parameter α and the scale parameter β_b depend on the original model [Bølviken, 2014, pp.325–326].

2.3.3.1 Determination of the Threshold

In practice the threshold is unknown and must be chosen. The choice of the threshold can sometimes be tricky. It has to be high enough so that the Pareto approach in the extreme right tail would be reliable. At the same time, due to few observations, it causes increased variance of the parameter estimates. The point is to find a balance between bias and variance. The traditional way of threshold determination is the so-called *fixed threshold approach*, i.e. the threshold is chosen before fitting the distribution. Commonly used methods for determining the threshold are *graphical diagnostics*. The user needs to graphically inspect the data and the model fit when determining the threshold. Some examples of graphical diagnostics could be the mean excess plot, the threshold stability plot or even more usual distribution fit diagnostics such as probability plots or quantile plots [Scarrott and MacDonald, 2012]. In this thesis, we will be using the `meanExcessPlot` from the `{laeken}` package in R.

2.3.3.2 Mixed Model with Pareto

Assume n claims z_1, \dots, z_n . By Pickand's theorem presented in section 2.3.3, above a chosen threshold b , $\mathbf{Z}_{>b}$ would be Pareto distributed whereas the under-threshold part, $\mathbf{Z}_{\leq b}$, would follow one of the distributions from section 2.3.1 or the empirical distribution from section 2.3.2. In other words, \mathbf{Z} is modelled by a mixture of two different distributions, either purely parametrically or semi-parametrically. Assume that the probability that \mathbf{Z} exceeds the threshold b is a small number ρ , i.e. $\Pr(\mathbf{Z} > b) = \rho$. A general algorithm of simulations from the mixed model is given below:

Algorithm 1 Simulating claims by mixed models

- 1: Input: Some threshold b chosen by `meanExcessPlot`, ρ , estimates of the parameters from some distribution in section 2.3.1 for $\mathbf{Z}_{\leq b}$, $\mathbf{Z}_{>b} - b \sim \text{Pareto}(\hat{\alpha}, \hat{\beta})$
 - 2: Draw $U^* \sim \text{Uniform}(0, 1)$
 - 3: **if** $U^* > \rho$ **then**
 - 4: $\mathbf{Z}^* \leftarrow$ Chosen distribution from section 2.3.1 % Central part
 else
 - 5: $\mathbf{Z}^* \leftarrow b + \text{Pareto}(\hat{\alpha}, \hat{\beta})$ % Extreme part
 - 6: Return \mathbf{Z}^* .
-

Extra command in line 1 is needed to sort the claims in ascending order $z_{(1)} \leq \dots \leq z_{(n)}$ if empirical distribution is fitted to $\mathbf{Z}_{\leq b}$ [Bølviken, 2014, pp.237–239].

2.4 Model for Portfolio Loss

In general insurance, the portfolio loss \mathcal{X} is defined as:

$$\mathcal{X} = Z_1 + \dots + Z_{\mathcal{N}}, \quad (2.52)$$

where \mathcal{N} and $Z_1, \dots, Z_{\mathcal{N}}$ are stochastically independent. Let $\mathbb{E}(Z_i) = \xi_z$ and $\text{sd}(Z_i) = \sigma_z$, where ξ_z and σ_z are estimated from one of the distributions presented in section 2.3. The mean and the variance of \mathcal{X} conditioned on \mathcal{N} are given by:

$$\begin{aligned} \mathbb{E}(\mathcal{X}|\mathcal{N}) &= \mathbb{E}(Z_1 + \dots + Z_{\mathcal{N}}|\mathcal{N}) = \sum_{i=1}^{\mathcal{N}} \mathbb{E}(Z_i) = \mathcal{N}\xi_z, \\ \text{Var}(\mathcal{X}|\mathcal{N}) &= \text{Var}(Z_1 + \dots + Z_{\mathcal{N}}|\mathcal{N}) = \sum_{i=1}^{\mathcal{N}} \text{Var}(Z_i) = \mathcal{N}\sigma_z^2. \end{aligned} \quad (2.53)$$

Hence, by the rules of double expectation and variance which are deduced in appendix A.1, the mean and the variance of \mathcal{X} are given by:

$$\mathbb{E}(\mathcal{X}) = \mathbb{E}(\mathcal{N})\xi_z \quad \text{and} \quad \text{Var}(\mathcal{X}) = \mathbb{E}(\mathcal{N})\sigma_z^2 + \text{Var}(\mathcal{N})\xi_z^2. \quad (2.54)$$

2.4.1 Uncertainty Evaluation Using Bootstrap

The total loss \mathcal{X} can be estimated through parameters from Z and \mathcal{N} . Since the parameters are estimated, the accuracy of the estimated parameters will affect the accuracy of the total losses. An often-used method for uncertainty evaluation is the bootstrap method, which can be used for instance for estimating the standard error of the estimators.

In general, the standard error of an estimator $\hat{\theta}$ is defined as the standard deviation of the sampling distribution $\sigma_{\hat{\theta}} = \sqrt{V(\hat{\theta})}$ [Devore and Berk, 2007, p.338]. It is essentially a measure of the accuracy of the estimates. Since the standard error itself needs to be estimated as well, it yields the *estimated standard error*, and is denoted by $s_{\hat{\theta}}$. Consider a random sample $\mathbf{x} = (x_1, x_2, \dots, x_n)$ from an unknown probability distribution F . Let $\theta = t(F)$ be the parameter of interest based on \mathbf{x} . The parameter is estimated by $\hat{\theta} = h(\mathbf{x})$, where $h(\cdot)$ is some function applied to \mathbf{x} . Now let \hat{F} be the empirical distribution with probability $1/n$ for each x_i . A bootstrap sample $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ is defined to be a random sample of n drawn from \hat{F} with replacement, which is the resampled version of the original \mathbf{x} . Under the bootstrap sample \mathbf{x}^* , a bootstrap replication of the $\hat{\theta}$:

$$\hat{\theta}^* = h(\mathbf{x}^*), \quad (2.55)$$

is produced. Hence, the bootstrap estimated standard error $\text{sd}_{\text{boot}}(\hat{\theta}^*)$ is namely the standard error of $\hat{\theta}^*$'s from the data sets \mathbf{x}^* based on \hat{F} [Efron and Tibshirani, 1994, pp.45–46]. Algorithm 2 below gives a more detailed description of the bootstrap procedure for estimating the standard error $\text{sd}_{\text{boot}}(\hat{\theta}^*)$ for the given data \mathbf{x} .

Algorithm 2 The bootstrap for estimating standard error

- 1: Choose m_b independent bootstrap samples $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_{m_b}^*$, where each bootstrap sample \mathbf{x}_i^* is drawn with replacement from \mathbf{x} .
- 2: From each bootstrap sample \mathbf{x}_i^* , obtain a bootstrap replication of the $\hat{\theta}$ by:

$$\hat{\theta}_i^* = h(\mathbf{x}_i^*), \text{ for } i = 1, \dots, m_b. \quad (2.56)$$

- 3: The bootstrap estimate of $\hat{\theta}$'s standard error is given by the sample standard deviation of the m_b replicates:

$$\text{sd}_{\text{boot}}(\hat{\theta}^*) = \sqrt{\frac{1}{m_b - 1} \sum_{i=1}^{m_b} (\hat{\theta}_i^* - \bar{\theta}^*)^2}, \quad (2.57)$$

where $\bar{\theta}^* = \sum_{i=1}^{m_b} \hat{\theta}_i^* / m_b$ [Efron and Tibshirani, 1994, p.47].

The bootstrap sampling can also be done parametrically. Unlike the \hat{F} that has been drawn empirically in the non-parametric bootstrap, \hat{F} is an estimate of F derived from a *parametric model*.

Since the bootstrap estimate of the standard error usually has relatively small bias, a small amount of replications is usually informative [Efron and Tibshirani, 1994, p.54]. However, the calculating capacity of the computers is much more powerful nowadays, a replication of $m_b = 500$ or 1000 is often used.

2.5 Reserve

The reserve, denoted as q_ϵ , is important in risk assessment and is also known as the solvency capital. The reserve ensures that the insurance and reinsurance companies are able to cover significant losses. It also ensures policyholders that payments will be made when they fall due [ii Association, 2016]. Mathematically, it is of the form:

$$\Pr(\mathcal{X} > q_\epsilon) = \epsilon, \quad (2.58)$$

where ϵ is a small number, and hence q_ϵ should be large enough to ensure that the probability that the total loss \mathcal{X} exceeds the amount of the reserve q_ϵ is very small. In other words, the reserve q_ϵ is the

upper ϵ -percentile of the portfolio liability \mathcal{X} .

As we are not able to get a mathematical expression for the reserve, the Monte Carlo simulation could be a smart solution for that. It will be explained in the section below.

2.5.1 Monte Carlo Simulation

The term *Monte Carlo* refers typically to the process of modelling and simulating of a system affected by randomness. The name "Monte Carlo" came from the famous Monte Carlo Casino in Monaco. The Monte Carlo simulation was first used in the 1950s at the Los Alamos laboratories for developing the hydrogen bomb [Thomopoulos, 2012, p.1]. Monte Carlo simulation is a good alternative when there is no explicit expression for the statistic we are interested in. Basically, the Monte Carlo simulation evaluates a statistic's behavior in random samples, and uses random samples from known populations of simulated data to track a statistic's behavior. [Mooney, 1997, p.2].

Suppose we want to estimate some quantity $\ell(\theta)$, where the quantity depends on some parameter θ . We assume that the expression for $\ell(\theta)$ is unknown. However, $\ell(\theta)$ can be written as:

$$\ell(\theta) = \int H(\mathbf{x})f(\mathbf{x}; \theta)dx = E(H(\mathbf{X})), \quad (2.59)$$

where \mathbf{X} is a random variable with pdf f depending on the parameter θ , and $H(\mathbf{x})$ is called the *performance* function. Then $\ell(\theta)$ can be estimated by the sample mean:

$$\hat{\ell} = \frac{1}{m} \sum_{i=1}^m H(\mathbf{X}_i), \quad (2.60)$$

where $\mathbf{X}_1, \dots, \mathbf{X}_m$ is a random sample from $f(x; \theta)$ [Rubinstein and Kroese, 2008, p.98]. The accuracy of the estimate depends on the number of simulations m . As $m \rightarrow \infty$, the estimate $\hat{\ell}$ should approach ℓ [Mahadevan, 1997, p.124].

In our case, the quantity ℓ is based on some parameter estimate $\hat{\theta}$, hence:

$$\hat{\ell}(\hat{\theta}) = \frac{1}{m} \sum_{i=1}^m H(\mathbf{X}_i^*), \quad (2.61)$$

where $\mathbf{X}_1^*, \dots, \mathbf{X}_m^*$ is simulated from $f(\mathbf{x}; \hat{\theta})$. As $m \rightarrow \infty$, $\hat{\ell}(\hat{\theta})$ should approach $\ell(\hat{\theta})$.

2.5.2 Estimation of the Reserve

Recall the model of the total loss from section 2.4. The total loss model based on all catastrophes is given by:

$$\begin{aligned} \mathcal{X}_{\text{Total}} &= Z_1^{\text{Total}} + \dots + Z_{\mathcal{N}_{\text{Total}}}^{\text{Total}}, \text{ where} \\ Z_j^{\text{Total}} &= Z_{\text{Type1},j} + Z_{\text{Type2\&3},j} + Z_{\text{Type4\&5\&9},j}. \end{aligned} \quad (2.62)$$

Similarly, the total loss model summed up by separate catastrophe types is given by:

$$\begin{aligned} \mathcal{X}_{\text{Type } i} &= Z_1^i + \dots + Z_{\mathcal{N}^i}^i \text{ for type } i, i = 1, 2\&3, 4\&5\&9, \\ \mathcal{X}_{\text{Separate}} &= \mathcal{X}_{\text{Type1}} + \mathcal{X}_{\text{Type2\&3}} + \mathcal{X}_{\text{Type4\&5\&9}}, \end{aligned} \quad (2.63)$$

where $\mathcal{N} = (N_1, \dots, N_j)$ and $\mathbf{Z} = (Z_1, \dots, Z_{\mathcal{N}})$. Now assume that the model for \mathcal{N} , $\mathbf{Z}_{\leq b}$ and $\mathbf{Z}_{> b}$ depend on the estimated parameters $(\hat{\mu}_{\text{Poisson}}, (\hat{\xi}_{\text{Gamma}}, \hat{\alpha}_{\text{Gamma}}))$ and $(\hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}})$, respectively. For convenience, denote $\hat{\theta} = (\hat{\mu}_{\text{Poisson}}, (\hat{\xi}_{\text{Gamma}}, \hat{\alpha}_{\text{Gamma}}), (\hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}}))$. We are looking for the estimate

$\hat{q}_\epsilon = q_\epsilon(\hat{\boldsymbol{\theta}})$. As mentioned in section 2.5.1, \hat{q}_ϵ has to be again estimated by the Monte Carlo simulation since we lack a closed form for that. The Monte Carlo simulated reserve based on the estimated parameters can be written as $\hat{q}_\epsilon^* = q_\epsilon^*(\hat{\boldsymbol{\theta}})$. An algorithm for the reserve estimate \hat{q}_ϵ^* is given below:

Algorithm 3 Monte Carlo simulation of the reserve

```

1: Input ML estimates  $\hat{\mu}_{\text{Poisson}}, \hat{\xi}_{\text{Gamma}}, \hat{\alpha}_{\text{Gamma}}, \hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}}, \rho, \epsilon$ 
2: for  $i = 1, \dots, m$  do
3:   Draw  $\hat{N}^* \sim \text{Poisson}(J\hat{\mu}_{\text{Poisson}}T)$ 
4:   for  $j = 1, \dots, \hat{N}^*$  do
5:     Draw  $U \sim \text{Uniform}(0, 1)$ 
6:     if  $U > \rho$  then
7:        $\hat{Z}_j^* \sim \text{Gamma}(\hat{\xi}_{\text{Gamma}}, \hat{\alpha}_{\text{Gamma}})$    % Central part
8:     else
9:        $\hat{Z}_j^* \leftarrow b + \text{Pareto}(\hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}})$    % Extreme part
10:     $\hat{\mathcal{X}}_i^* = \sum_{j=1}^{\hat{N}^*} \hat{Z}_j^*$ 
11: Sort  $\hat{\mathcal{X}}_1^*, \dots, \hat{\mathcal{X}}_m^*$  as  $\hat{\mathcal{X}}_{(1)}^* \leq, \dots, \leq \hat{\mathcal{X}}_{(m)}^*$ 
12: Return  $\hat{q}_\epsilon^* \leftarrow \hat{\mathcal{X}}_{m(1-\epsilon)}^*$ 

```

Line 3 and line 7 in algorithm 3 might be replaced by one of the other distributions presented in section 2.2 and 2.3.

Uncertainty is inevitable when estimation is involved. In our case, the possible errors involved are as follows:

$$\begin{aligned}
\hat{q}_\epsilon^* - q_\epsilon(\boldsymbol{\theta}) &\longrightarrow \text{Total error} \\
\hat{q}_\epsilon^* - \hat{q}_\epsilon &\longrightarrow \text{Monte Carlo error} \\
\hat{q}_\epsilon - q_\epsilon(\boldsymbol{\theta}) &\longrightarrow \text{Estimation error.}
\end{aligned}$$

Hence the total error can be written as:

$$\hat{q}_\epsilon^* - q_\epsilon(\boldsymbol{\theta}) = \hat{q}_\epsilon^* - \hat{q}_\epsilon + \hat{q}_\epsilon - q_\epsilon(\boldsymbol{\theta}). \quad (2.64)$$

Theoretically, $\hat{q}_\epsilon^* \rightarrow \hat{q}_\epsilon$ as $m \rightarrow \infty$, and $\hat{q}_\epsilon \rightarrow q_\epsilon(\boldsymbol{\theta})$ as $n \rightarrow \infty$, where m, n are the number of Monte Carlo simulations and the number of observations from historical data, respectively. However, the number of observations n is not something under our control, and it really depends on the data we got, and we have very limited ability to influence the data. Also, the severity of the estimation error is usually larger than the Monte Carlo error [Bølviken, 2014, pp.230–232].

Another possible source of error which is not mentioned above is the model bias, i.e. the uncertainty of if there exists another model that fits the data even better than the one we assumed. This type of error is difficult to quantify. Fortunately, the model bias usually does not count that much in comparison with the Monte Carlo and the estimation error, if a reasonable model is chosen.

As presented in section 2.4.1, bootstrap method is often used for parameter uncertainty evaluation. Along with the Monte Carlo simulation for \hat{q}_ϵ^* , a nested bootstrap Monte Carlo algorithm is required. A more explicit algorithm is given below:

Algorithm 4 The Poisson/Gamma/Pareto reserve bootstrap

```
1: Input ML estimates  $\hat{\mu}_{\text{Poisson}}, \hat{\xi}_{\text{Gamma}}, \hat{\alpha}_{\text{Gamma}}, \hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}}$  and  $A, J, T, \rho, \epsilon$ 
2: Draw  $\hat{\mathcal{N}}^* \sim \text{Poisson}(A\hat{\mu}_{\text{Poisson}})$ ,  $\hat{\mu}_{\text{Poisson}}^* \leftarrow \hat{\mathcal{N}}^*$  % Bootstrap estimates
3: for  $i = 1, \dots, m_b$  do
4:   Draw  $U_1 \sim \text{Uniform}(0, 1)$ 
5:   if  $U_1 > \rho$  then
6:      $\hat{Z}_i^* \sim \text{Gamma}(\hat{\xi}_{\text{Gamma}}, \hat{\alpha}_{\text{Gamma}})$ 
7:   else
8:      $\hat{Z}_i^* \sim \text{Pareto}(\hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}})$ 
9:    $(\hat{\xi}_{\text{Gamma}}^*, \hat{\alpha}_{\text{Gamma}}^*) \xleftarrow{ML} \hat{Z}_{\leq b}^*$ 
    $(\hat{\alpha}_{\text{Pareto}}^*, \hat{\beta}_{\text{Pareto}}^*) \xleftarrow{ML} \hat{Z}_{> b}^*$ 
10:  $\hat{\mathcal{N}}^{**} \sim \text{Poisson}(J\hat{\mu}_{\text{Poisson}}^*)$ 
11: for  $j = 1, \dots, m$  do
12:   Draw  $U_2 \sim \text{Uniform}(0, 1)$ 
13:   if  $U_2 > \rho$  then
14:      $\hat{Z}_j^{**} \sim \text{Gamma}(\hat{\xi}_{\text{Gamma}}^*, \hat{\alpha}_{\text{Gamma}}^*)$ 
15:   else
16:      $\hat{Z}_j^{**} \leftarrow b + \text{Pareto}(\hat{\alpha}_{\text{Pareto}}^*, \hat{\beta}_{\text{Pareto}}^*)$ 
17:    $\hat{\mathcal{X}}_j^{**} \leftarrow \sum_{j=1}^{\hat{\mathcal{N}}^{**}} \hat{Z}_j^{**}$ 
18: Sort  $\hat{\mathcal{X}}_1^{**}, \dots, \hat{\mathcal{X}}_m^{**}$  as  $\hat{\mathcal{X}}_{(1)}^{**} \leq \dots \leq \hat{\mathcal{X}}_{(m)}^{**}$ 
19: Return  $\hat{q}_\epsilon^{**} \leftarrow \hat{\mathcal{X}}_{(m(1-\epsilon))}^{**}$ .
```

Again, line 2, 6, 9, 10 and 14 in algorithm 4 might be adjusted accordingly depending on the distribution fitted to the data. The total amount of simulations is $m \times m_b$. The number of bootstrap replication follows the rule in section 2.4.1.

Chapter 3

Data Analysis

In this chapter, distributions presented in the previous chapter will be applied to the data. Description of the data will be given first. The modelling of the catastrophe frequency and the amounts of compensation will be separated into two sections. In the first section, data will be modelled all together. In the second section, data will be modelled separately based on catastrophe types. For the catastrophe frequency, the Poisson and the negative binomial distribution will be fitted to the data. For the amounts of compensation, since extreme situations may occur, a threshold will first be chosen using the `meanExcessPlot`. To the data below the threshold, the log-normal, Gamma, Pareto and extended Pareto distribution will be fitted. In the extreme right tail, the Pareto will be fitted as default to the data. All models are fitted by the statistical software R.

3.1 Data

Data utilized in this thesis is based on the natural disasters recorded between the year 1980 and 2013 from Finans Norge. The data set consists of seven columns categorized by the date of occurrence, municipality code, catastrophe type, event number of which the catastrophe is related to and amounts of compensation. The event number column is numbered from 0 to 55, depending on which case the catastrophes are related to. Catastrophes that are not related to any particular events are numbered with 0. The catastrophe types are sorted by storm, storm surge, flood, avalanche, earthquake, volcanic eruptions and other/unknown with number 1-6 and 9, respectively. No occurrence of volcanic eruptions was recorded in the data.

The catastrophes that happened in the same year and same places of the same type which corresponds to the third, fourth and fifth column in the data set are aggregated into the claim numbers N_{jk}^i , organized in a 3-dimensional array, representing the municipality with index j , the year with index k and the catastrophe type with index i , respectively. The number of the municipalities goes from $j = 1, \dots, J$ with $J = 431$. The index of the year goes from $k = 1, \dots, K$ with $K = 34$ and the index of the catastrophe type is given by $i = 1, \dots, I$ with $I = 6$.

When modelling the amounts of compensation, the last column can be extracted directly by specifying the catastrophe type from the fifth column of the data set. The amounts of compensation that are less or equal than 3 are removed from the data set before modelling since they somehow cause strange behaviour in the models.

3.2 Common Model for All Catastrophe Types

3.2.1 Claim Frequency Modelling

- Poisson distribution:

Since we have a common model for all catastrophe types, we need to sum up all i -indexes for N_{jk}^i defined above. By (2.12), the estimate of the parameter μ_{total} can be calculated by

$$\hat{\mu}_{\text{total}} = \frac{\text{Total number of claims}}{\text{Total risk exposure A}}. \quad (3.1)$$

As explained in section 2.2.1, since catastrophes may occur in all places during all years, the total risk exposure is $J \times K \times T$ for $J = 431$, $K = 34$ and $T = 1$. Hence, by (3.1):

$$\hat{\mu}_{\text{total}} = \frac{84876}{431 \times 34 \times 1} = 5.79.$$

The accuracy of the distribution can be checked through for instance a QQ-plot and a density plot. A straight line in the QQ-plot and a coincidence of the density plots indicate a good fit. Figure 3.1 shows the QQ-plot and the density plot of the Poisson distribution in comparison with the actual data. The QQ-plot does not produce a straight line, and the density plot does not coincide that much with the actual data, indicating that the Poisson distribution is not a proper model for the given data.

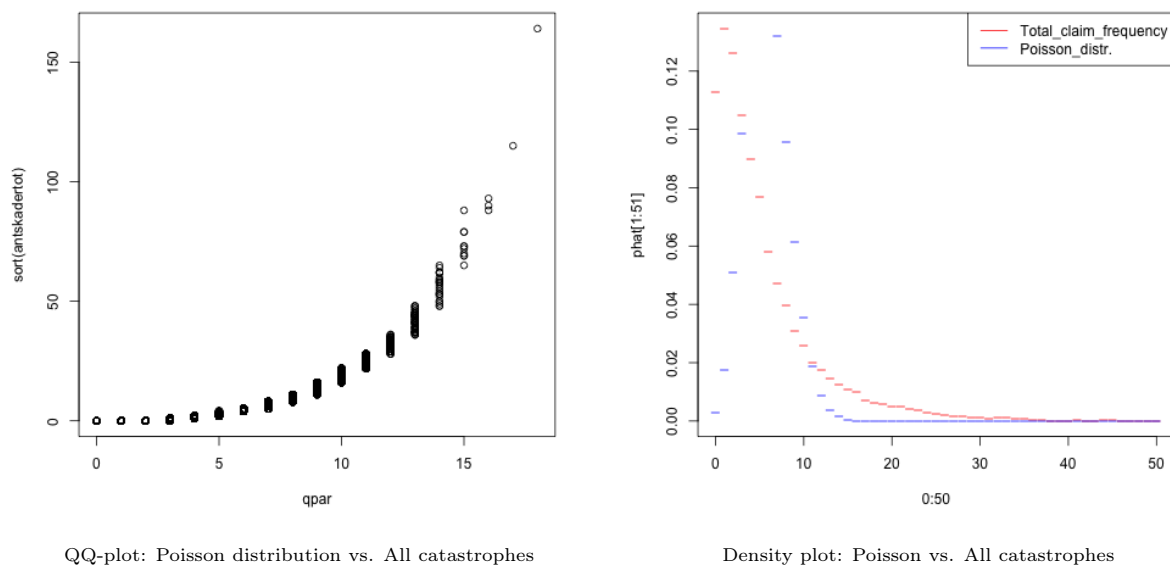
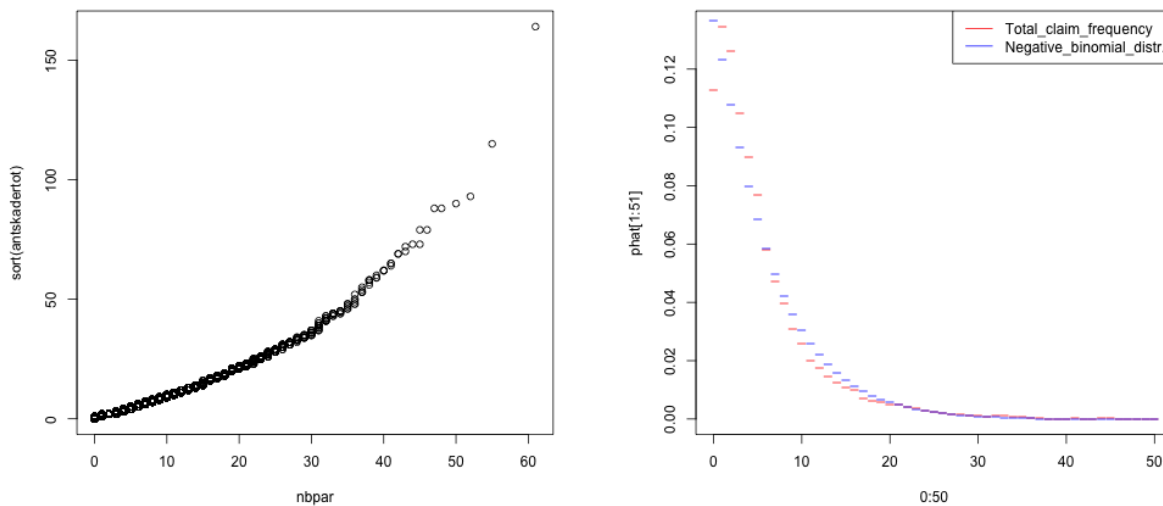


Figure 3.1: Fitting of the catastrophe frequency by the Poisson distribution

- Negative binomial distribution:

The parameters ξ_{total} and α_{total} are estimated numerically by the maximum likelihood method in (2.23), the estimates of the parameters are given by:

$$\hat{\xi}_{\text{total}} = 5.79 \quad \text{and} \quad \hat{\alpha}_{\text{total}} = 1.07.$$



QQ-plot: NB distribution vs. All catastrophes

Density plot: NB vs. All catastrophes

Figure 3.2: Fitting of the catastrophe frequency by the negative binomial distribution

Comparing figure 3.1 from the Poisson distribution and figure 3.2 from the negative binomial distribution, we observe that the negative binomial distribution is a better model for the data since the QQ-plot yields a straighter line and the density plot almost overlaps the actual data. The plots indicate that the negative binomial distribution is rather a good model.

3.2.2 Claim Size Modelling

As mentioned in section 2.3.3.1, the fixed threshold approach is used for the determination of the threshold. The choice of the threshold can be decided by the "meanExcessPlot" in R, which is an often-used graphic method for detecting the threshold of a Pareto distribution. The mean excesses are plotted against all claim sizes, and the tail of the plot will be almost linear if the data follows a Pareto distribution. Therefore the starting point of a straight line would be a good choice for the threshold.

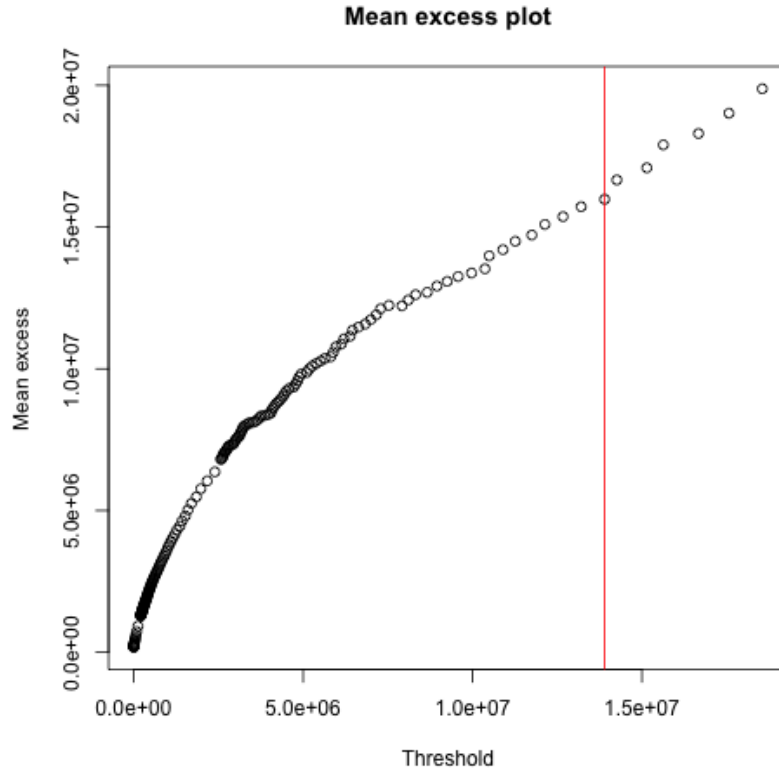


Figure 3.3: The mean excess plot for all claim sizes

By testing several different values on the x-axis, we observe that claim size 13898075 can be a good threshold for the data (see figure 3.3), i.e. all data over this threshold would follow a Pareto distribution according to the Pickand's theorem. By the `ecdf` function we find that the percentile below the threshold is approximately 99.8%, which corresponds to an upper 0.2% percentile of the data.

Central part:

- Log-normal distribution:

By (2.30) the estimates of the mean and the standard deviation of $\log(Z)$ are given by:

$$\bar{y}_{\text{total}} = 9.96 \quad \text{and} \quad s_{y_{\text{total}}} = 1.77.$$

Hence, by (2.32) the estimates of the parameters from Z are given by:

$$\hat{\xi}_{\text{total}} = 1.01 \cdot 10^5 \quad \text{and} \quad \hat{\sigma}_{\text{total}} = 1.77.$$

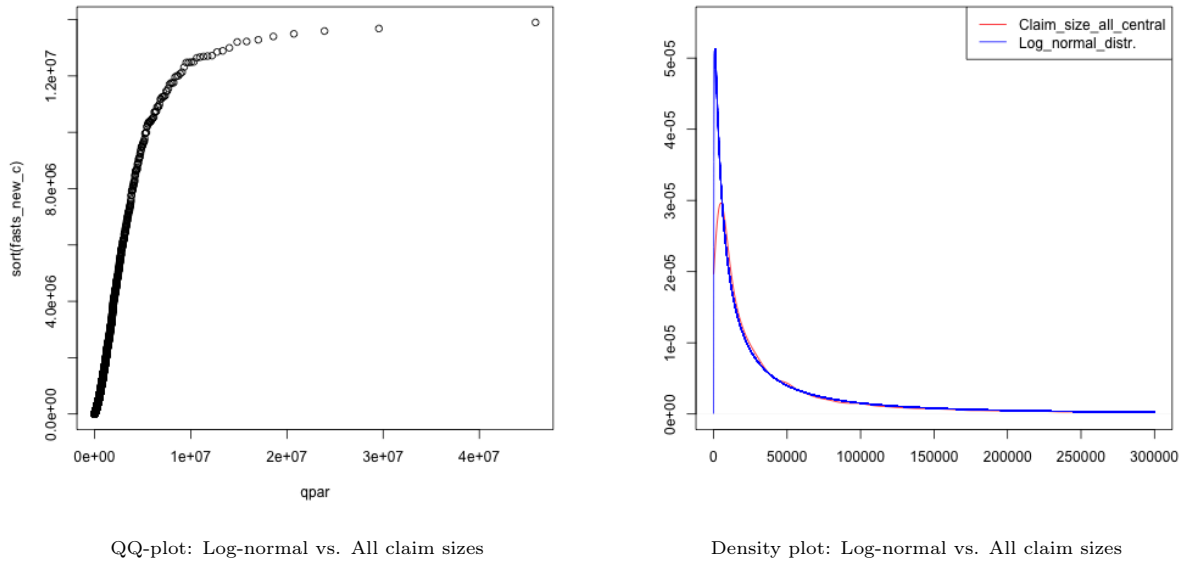


Figure 3.4: Fitting of all claim sizes by the log-normal distribution

From the QQ- and the density plots in figure 3.4, we see that the log-normal distribution is not a good model for the data.

- Gamma distribution:

By the MLE of ξ given in (2.35) and by optimizing $\hat{\alpha}$ in (2.36) numerically, the estimates of the parameters are given by:

$$\hat{\xi}_{\text{total}} = 1.32 \cdot 10^5 \quad \text{and} \quad \hat{\alpha}_{\text{total}} = 0.36.$$

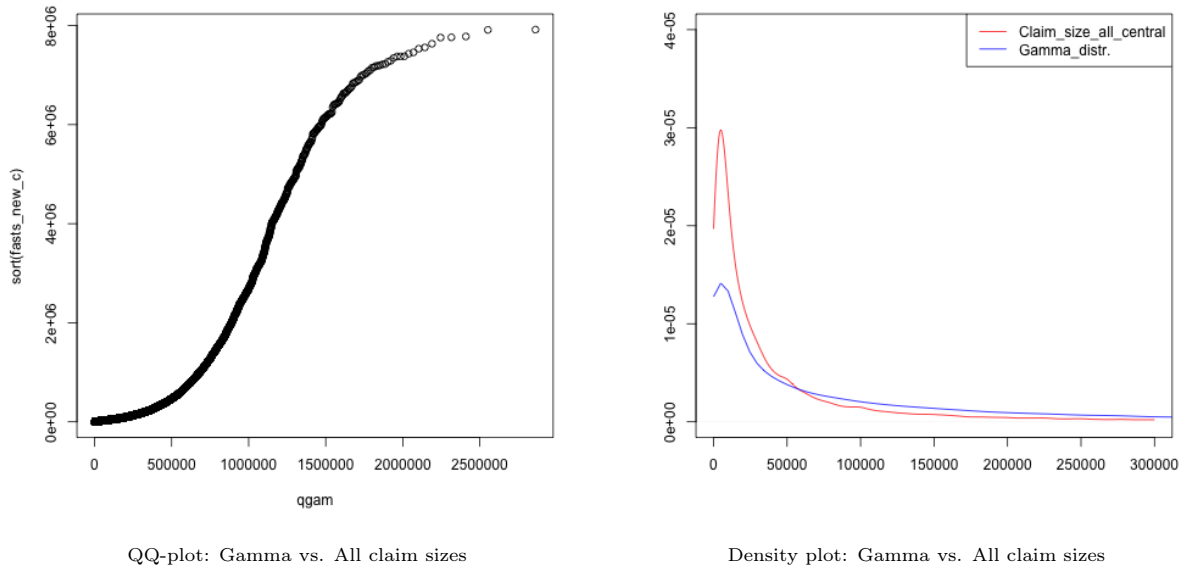


Figure 3.5: Fitting of all claim sizes by the Gamma distribution

Compared to the log-normal distribution, there is a slight improvement we can see in the QQ-plot and the density plot in figure 3.5, but the fit is still not good.

When the distribution does not fit the data well, the data could be transformed with the help of some monotone, continuous functions so that the distribution can fit the transformed data better than the original one. Here we try transforming the data by taking the logarithm, and assume that:

$$\log(Z) \sim \text{Gamma}(\xi_{\text{total,log}}, \alpha_{\text{total,log}}). \quad (3.2)$$

With the same procedure, the estimates are now given by:

$$\hat{\xi}_{\text{total,log}} = 9.96 \quad \text{and} \quad \hat{\alpha}_{\text{total,log}} = 31.26.$$

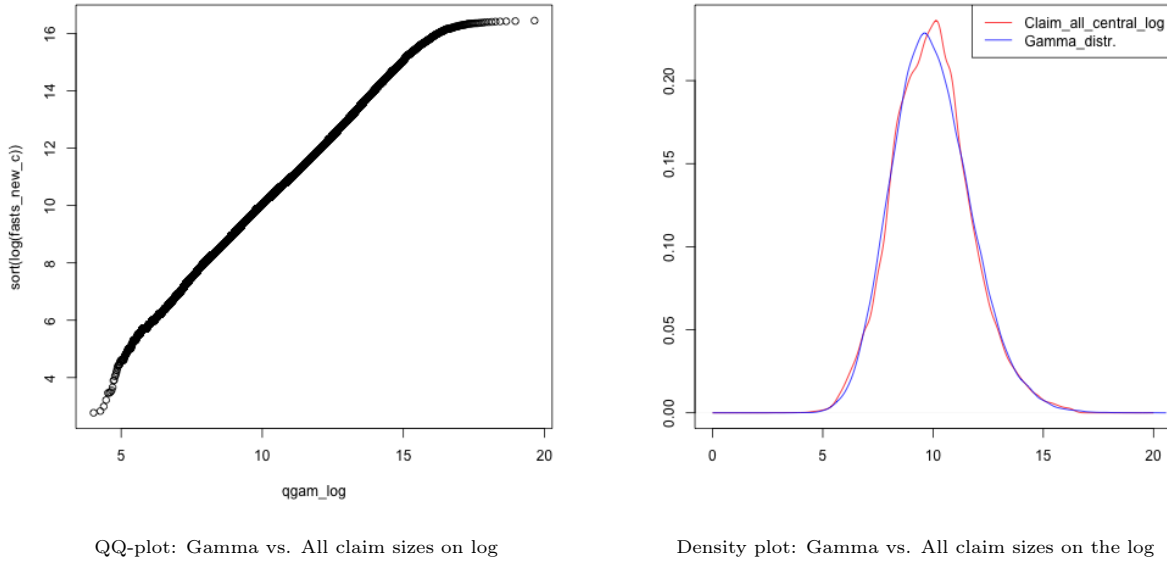


Figure 3.6: Fitting of all claim sizes on the log scale by the Gamma distribution

In figure 3.6 we see a huge improvement after the data transformation, there is almost a linear relation in the QQ-plot, and the density plot overlaps with the actual data as well. Therefore the log-Gamma distribution is a very good model.

- Pareto distribution:

By optimizing the log-likelihood function in (2.42) numerically we get the estimate of β_{total} :

$$\hat{\beta}_{\text{total}} = 9.11 \cdot 10^{19}.$$

Inserting $\hat{\beta}_{\text{total}}$ into the expression for $\hat{\alpha}_{\beta}$ in (2.41), the estimate for $\alpha_{\beta_{\text{total}}}$ is given by:

$$\hat{\alpha}_{\beta_{\text{total}}} = 6.96 \cdot 10^{14}.$$

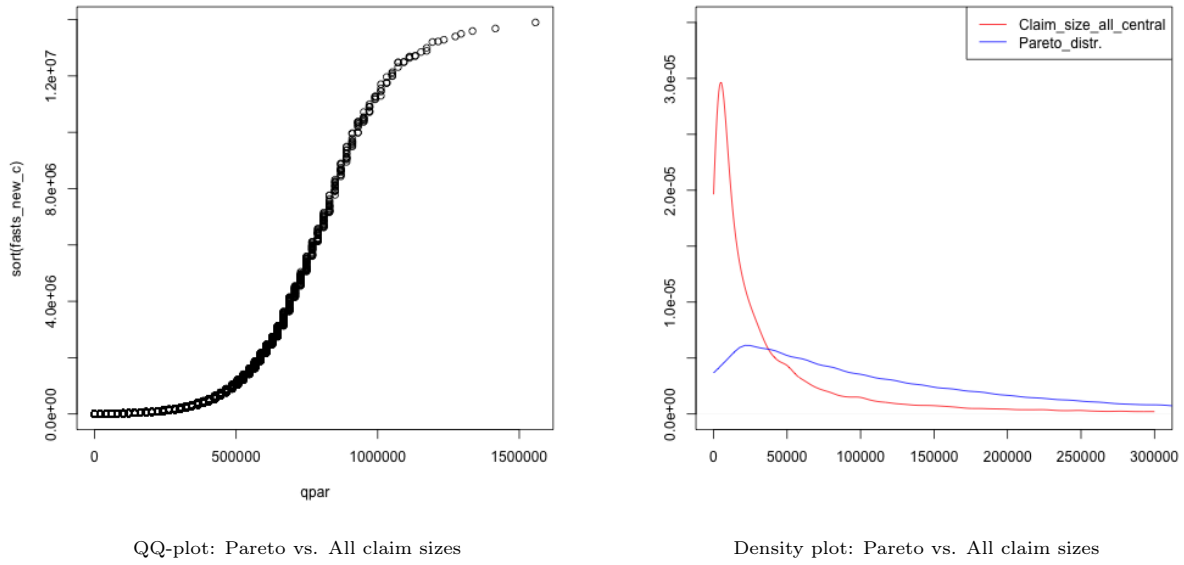


Figure 3.7: Fitting of all claim sizes by the Pareto distribution

According to figure 3.7, the Pareto distribution is not a good model for the data. We try now the generalized Pareto, i.e. the extended Pareto.

- Extended Pareto distribution:

Optimizing the log-likelihood function in (2.46), the estimates of the parameters are given by:

$$\hat{\alpha}_{\text{total}} = 0.86, \quad \hat{\beta}_{\text{total}} = 1.13 \cdot 10^4 \quad \text{and} \quad \hat{\theta}_{\text{total}} = 1.26.$$

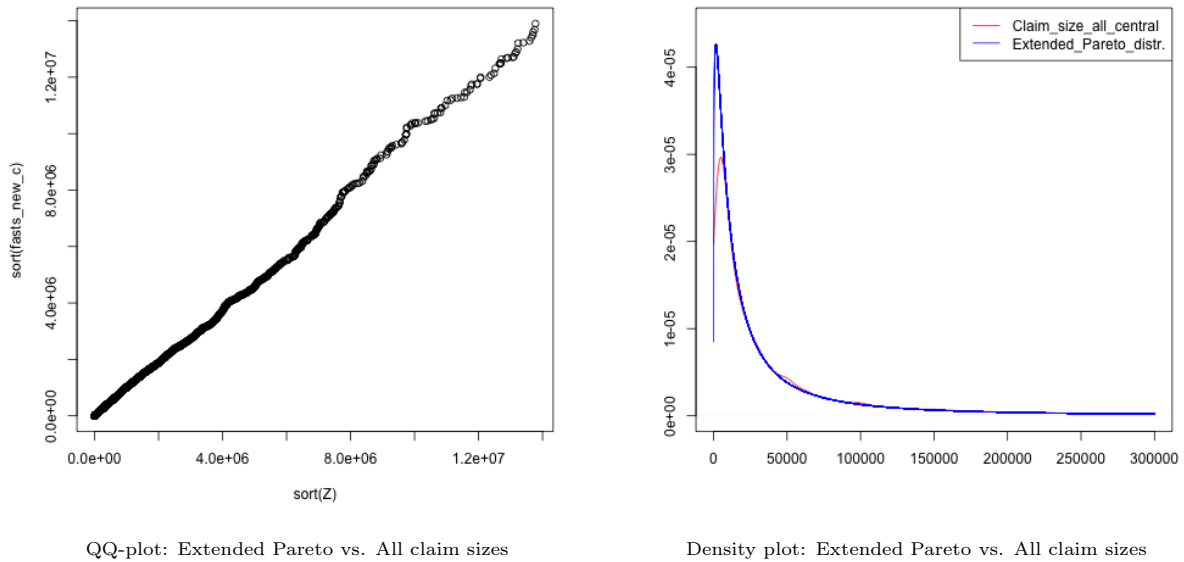


Figure 3.8: Fitting of all claim sizes by the Extended Pareto distribution

Compared with the Pareto distribution, the extended Pareto in figure 3.8 fits better due to the one more

parameter. It is not a bad model for the data since the QQ-plot yields almost a straight line. However, we prefer the log-normal distribution as it has one less parameter.

Over-threshold part:

The over-threshold part data consists of a total of 113 observation. According to the extreme value theory, with an ideal threshold, the extreme right tail of the data should follow the Pareto distribution. With the same procedure as in the central part, the estimates of the Pareto parameters are given by:

$$\hat{\beta}_{\text{over}} = 1.94 \cdot 10^7 \quad \text{and} \quad \hat{\alpha}_{\beta_{\text{over}}} = 2.12.$$

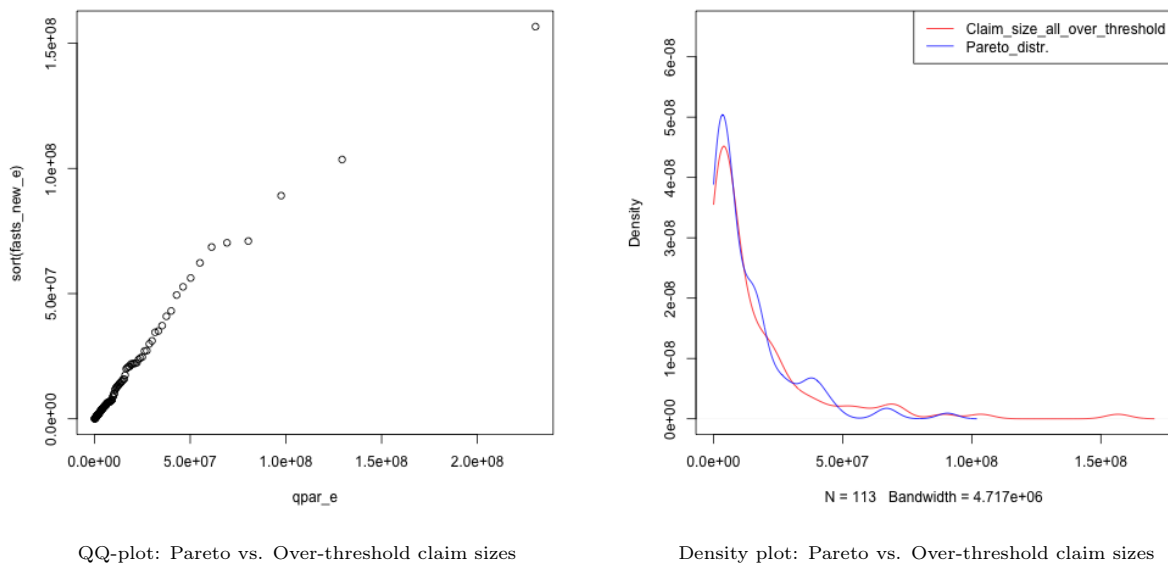


Figure 3.9: Fitting of the over-threshold claim sizes by the Pareto distribution

Based on the chosen threshold in figure 3.3, the Pareto distribution fits the extreme data quite well according to figure 3.9.

3.3 Separate Models for Different Catastrophe Types

Since the catastrophe frequency and the amounts of compensation may vary depending on the type, they will be modelled separately. Except for catastrophe type 1, there are relatively fewer data of other types recorded. Therefore we consider merging the data of similar catastrophe types. The catastrophe type 2 and type 3 both belong to the flood category, therefore we put the data of these two types together. The remaining three catastrophe types are more unusual, so they are put together as a single type.

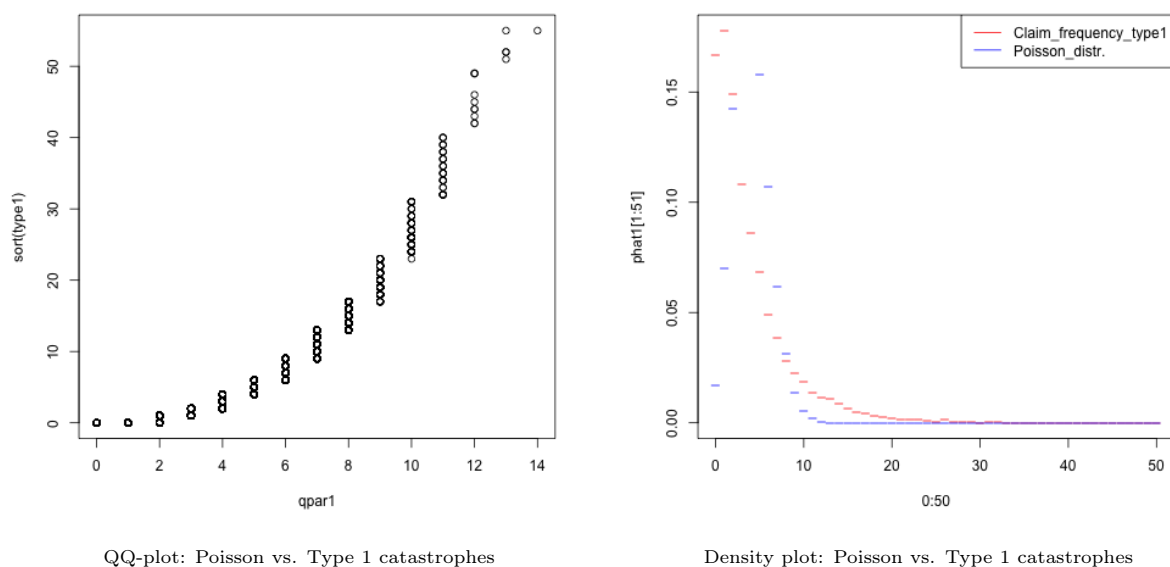
3.3.1 Catastrophe Type 1

3.3.1.1 Claim Frequency Modelling

- Poisson distribution:

By the same procedure as in the previous section, the estimate of the parameter μ_1 is given by:

$$\begin{aligned}\hat{\mu}_1 &= \frac{\text{Claim numbers of type 1}}{A} = \frac{\text{Claim numbers of type 1}}{J \times K \times T} \\ &= \frac{59437}{431 \times 34 \times 1} \\ &= 4.06.\end{aligned}$$



QQ-plot: Poisson vs. Type 1 catastrophes

Density plot: Poisson vs. Type 1 catastrophes

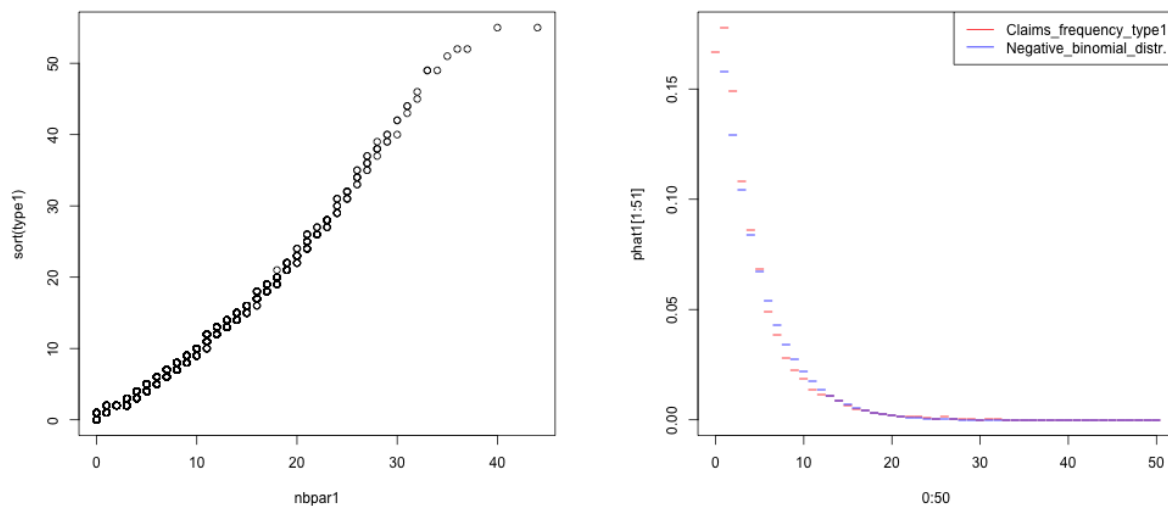
Figure 3.10: Fitting of the catastrophe frequency of type 1 by the Poisson distribution

According to figure 3.10, the Poisson distribution is not a good enough model for the number of catastrophes of type 1.

- Negative binomial distribution:

The estimates of parameters ξ_1 and α_1 are given by:

$$\hat{\xi}_1 = 4.06 \quad \text{and} \quad \hat{\alpha}_1 = 1.06. \quad (3.3)$$



QQ-plot: NB vs. Type 1 catastrophes

Density plot: NB vs. Type 1 catastrophes

Figure 3.11: Fitting of the catastrophe frequency of type 1 by the negative binomial distribution

According to figure 3.11, the negative binomial distribution describes the catastrophe numbers of type 1 very well.

3.3.1.2 Claim Size Modelling

From the meanExcessPlot in figure 3.12 we observe that right to the claim size of 6903315 it forms almost a straight line. Therefore it can be a good threshold for the claim size of catastrophes of type 1. The over-threshold part data corresponds to an upper 0.3% percentile of the data.

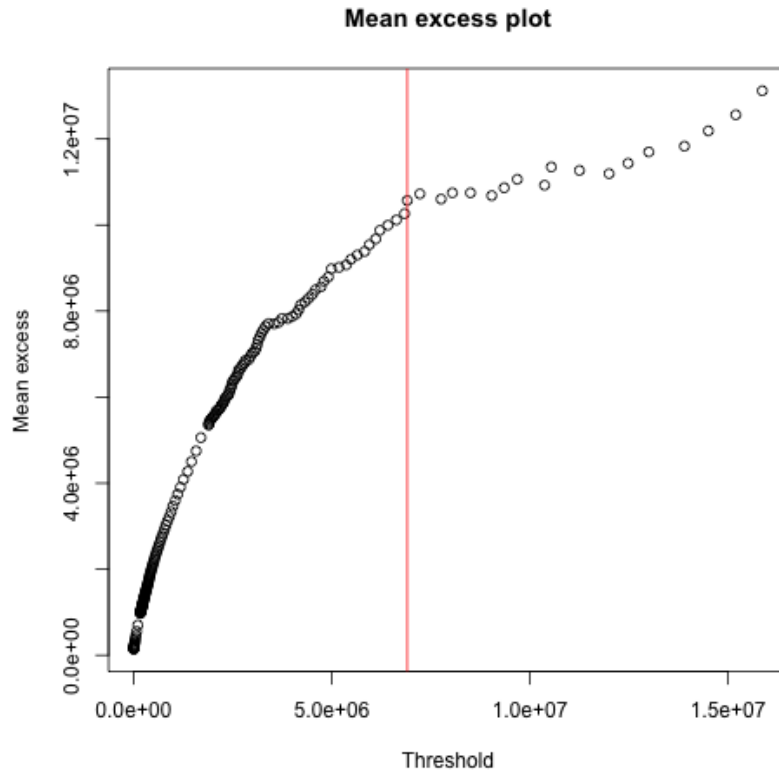


Figure 3.12: The mean excess plot for type 1 claim sizes

Central part:

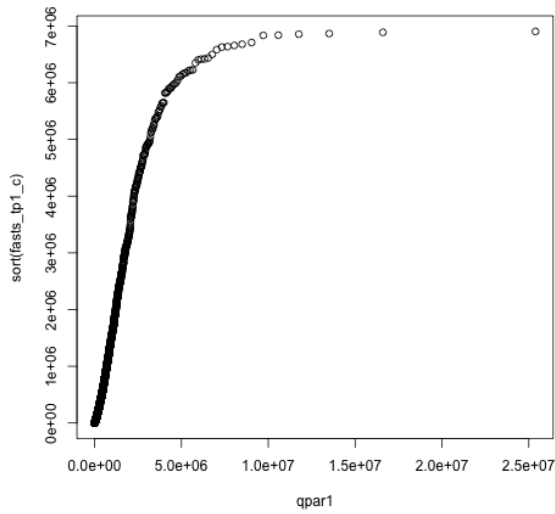
- Log-normal distribution:

The estimates of the mean and the standard deviation of $\log(Z)$ are given by:

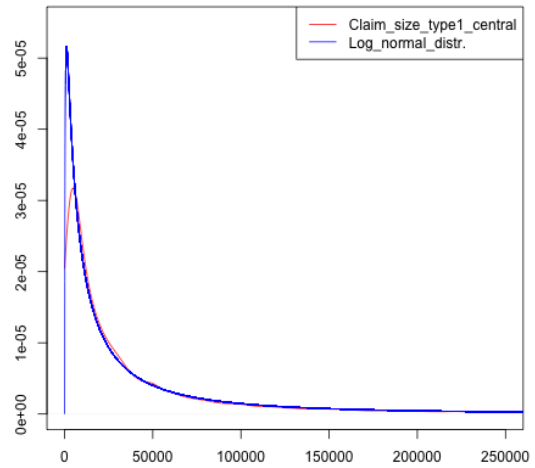
$$\bar{y}_1 = 9.85 \quad \text{and} \quad s_{y_1} = 1.69.$$

Hence the estimates of the parameters of Z are given by:

$$\hat{\xi}_1 = 7.88 \cdot 10^4 \quad \text{and} \quad \hat{\sigma}_1 = 1.69.$$



QQ-plot: Log-normal vs. Type 1 claim sizes



Density plot: Log-normal vs. Type 1 claim sizes

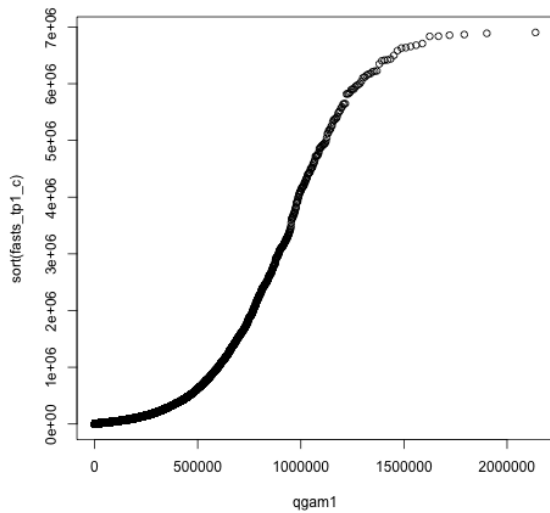
Figure 3.13: Fitting of the claim sizes of type 1 by the log-normal distribution

According to figure 3.13, the log-normal distribution gives a bad fit to the type 1 data.

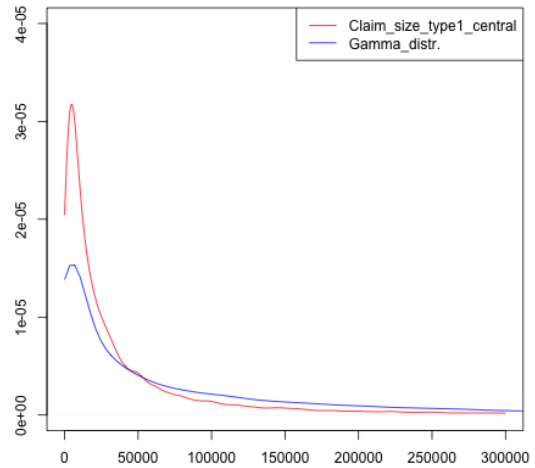
- Gamma distribution:

By the maximum likelihood method, the estimates of the parameters are given by:

$$\hat{\xi}_1 = 9.36 \cdot 10^4 \quad \text{and} \quad \hat{\alpha}_1 = 0.41. \quad (3.4)$$



QQ-plot: Gamma vs. Type 1 claim sizes



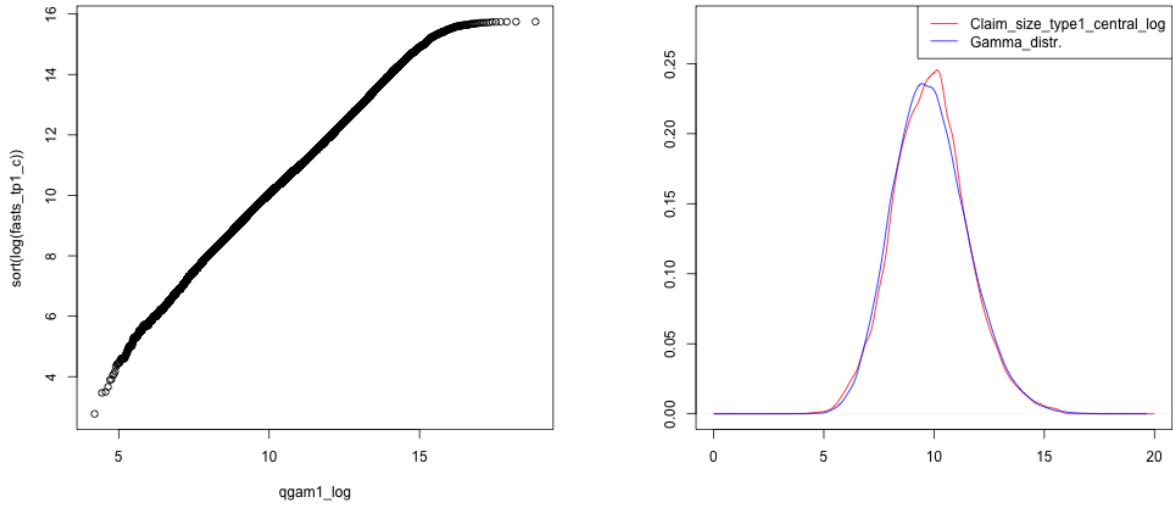
Density plot: Gamma vs. Type 1 claim sizes

Figure 3.14: Fitting of the claim sizes of type 1 by the Gamma distribution

Observe from figure 3.14 that the ordinary Gamma distribution does not fit the claim sizes of catastrophe type 1 well. Therefore, we try fitting the Gamma distribution to the logarithm of the data. The estimates

are given by:

$$\hat{\xi}_{1,\log} = 9.85 \quad \text{and} \quad \hat{\alpha}_{1,\log} = 33.54. \quad (3.5)$$



QQ-plot: Gamma vs. Type 1 claim sizes on the log

Density plot: Gamma vs. Type 1 claim sizes on the log

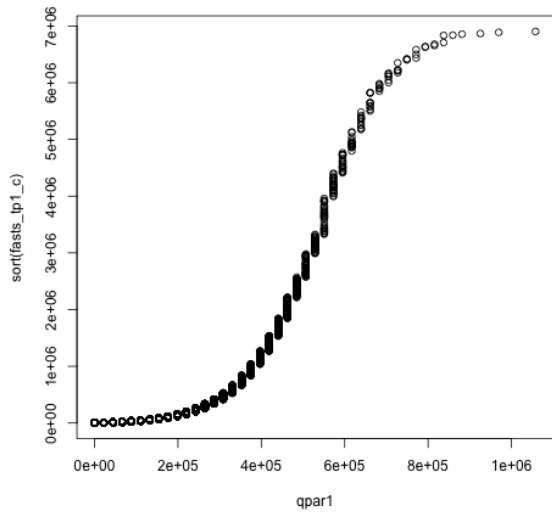
Figure 3.15: Fitting of the claim sizes of type 1 on the log scale by the Gamma distribution

Figure 3.15 above shows that the log-Gamma distribution is rather a good model for the claim sizes of catastrophe type 1.

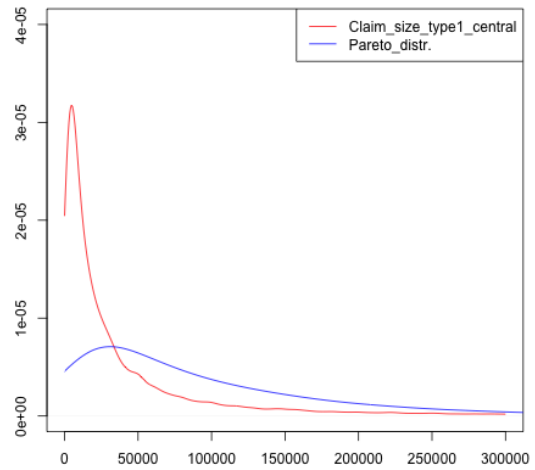
- Pareto distribution:

Using the maximum likelihood method, the estimates of the parameters β_1 and α_{β_1} are given by:

$$\hat{\beta}_1 = 9.92 \cdot 10^{19} \quad \text{and} \quad \hat{\alpha}_{\beta_1} = 1.07 \cdot 10^{15}.$$



QQ-plot: Pareto vs. Type 1 claim sizes



Density plot: Pareto vs. Type 1 claim sizes

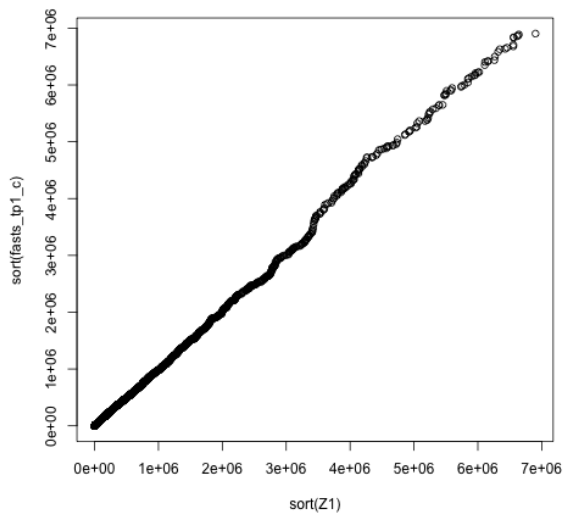
Figure 3.16: Fitting of the claim sizes of type 1 by the Pareto distribution

The Pareto distribution does not fit the under-threshold part of the type 1 claim sizes well according to 3.16.

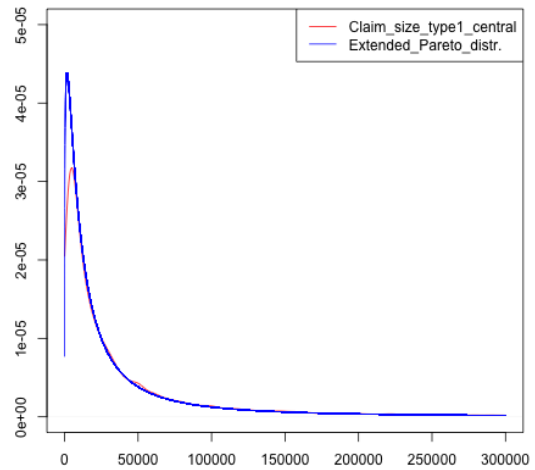
- Extended Pareto distribution:

By the maximum likelihood method, the estimates of the parameters are given by:

$$\hat{\alpha}_1 = 0.95, \quad \hat{\beta}_1 = 1.19 \cdot 10^4 \quad \text{and} \quad \hat{\theta}_1 = 1.27.$$



QQ-plot: Extended Pareto vs. Type 1 claim sizes



Density plot: Extended Pareto vs. Type 1 claim sizes

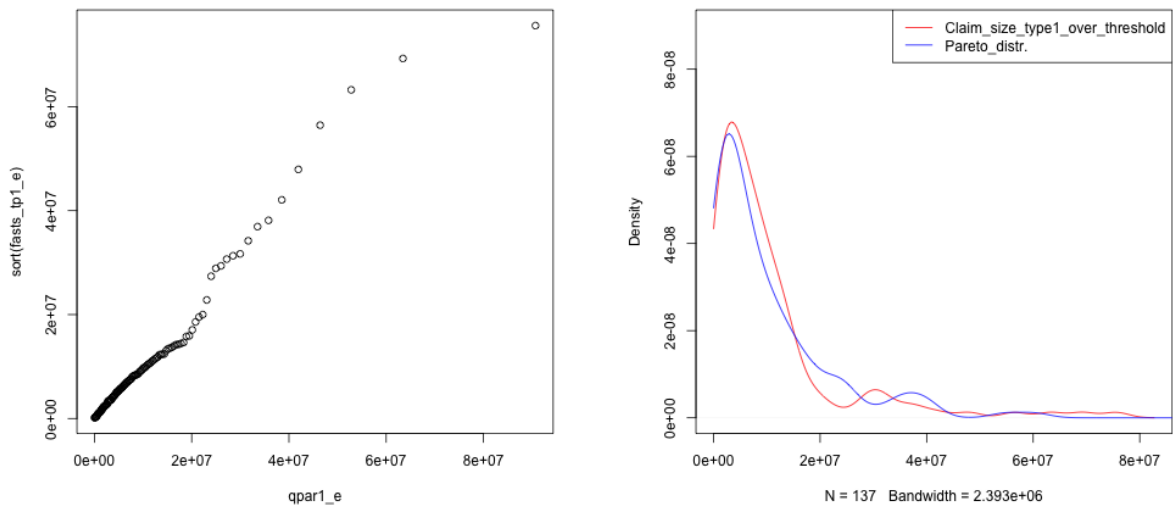
Figure 3.17: Fitting of the claim sizes of type 1 by the Extended Pareto distribution

Figure 3.17 indicates that the extended Pareto distribution gives an improvement in comparison with the Pareto. Again, we prefer the log-Gamma distribution because it has fewer parameters.

Over-threshold part:

The over-threshold part data consists of a total of 137 observations. The estimates of the Pareto parameters are given by:

$$\hat{\beta}_{1,\text{over}} = 3.85 \cdot 10^7 \quad \text{and} \quad \hat{\alpha}_{\beta_{1,\text{over}}} = 4.64.$$



QQ-plot: Pareto vs. Type 1 over-threshold claim sizes

Density plot: Pareto vs. Type 1 over-threshold claim sizes

Figure 3.18: Fitting of the over-threshold claim sizes of type 1 by the Pareto distribution

Above the threshold we have chosen, the Pareto distribution fits the extreme right tail of the type 1 claim sizes quite well as figure 3.18 shown.

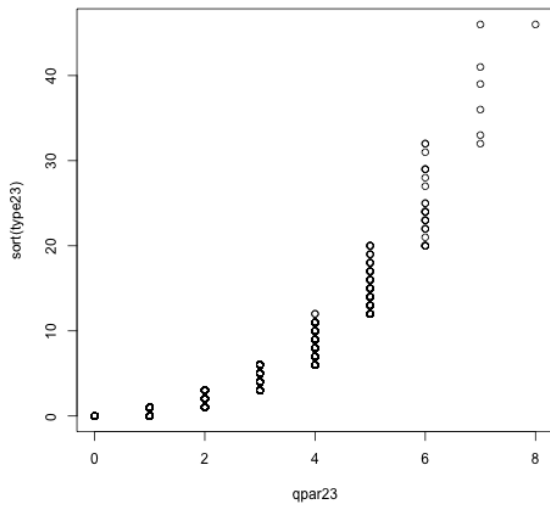
3.3.2 Catastrophe Type 2 and Type 3

3.3.2.1 Claim Frequency Modelling

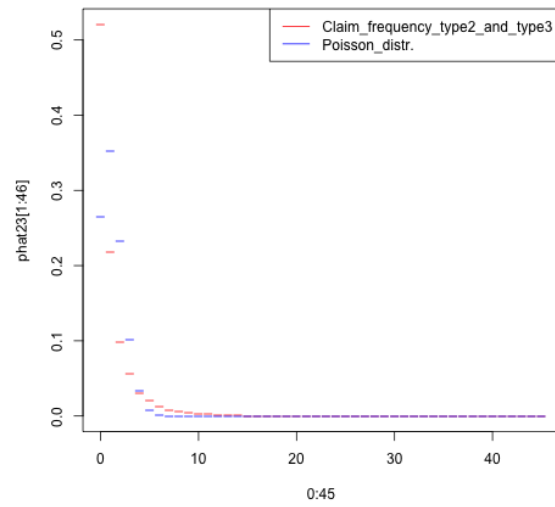
- Poisson distribution:

The estimate of the parameter $\mu_{2\&3}$ is given by:

$$\hat{\mu}_{2\&3} = \frac{\text{Claim numbers of type 2 and type 3}}{A} = \frac{19396}{431 \times 34} = 1.32.$$



QQ-plot: Poisson vs. Type 2 and 3 catastrophes



Density plot: Poisson vs. Type 2 and 3 catastrophes

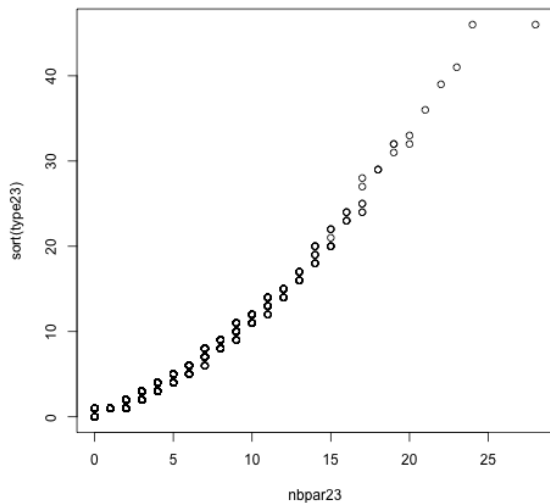
Figure 3.19: Fitting of the catastrophe frequency of type 2 and type 3 by the Poisson distribution

For the number of catastrophes of type 2 and 3, according to figure 3.19, the Poisson distribution does not fit the data very well.

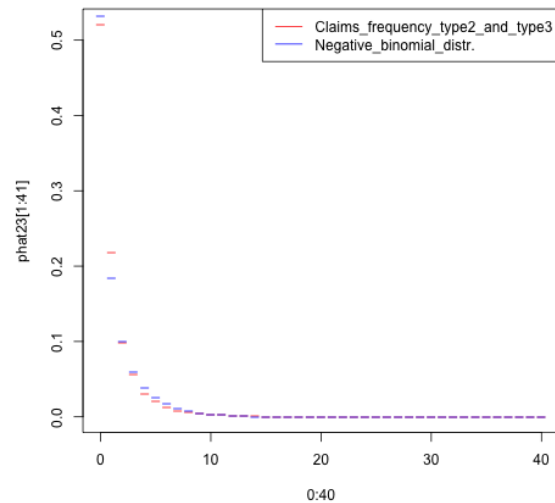
- Negative binomial distribution:

The estimates of parameters $\xi_{2\&3}$ and $\alpha_{2\&3}$ are given by:

$$\hat{\xi}_{2\&3} = 1.32 \quad \text{and} \quad \hat{\alpha}_{2\&3} = 0.47.$$



QQ-plot: NB vs. Type 2 and 3 catastrophes



Density plot: NB vs. Type 2 and 3 catastrophes

Figure 3.20: Fitting of the catastrophe frequency of type 2 and type 3 by the negative binomial distribution

As figure 3.20 shown above, the negative binomial distribution describes the number of catastrophes of type 2 and 3 very well.

3.3.2.2 Claim Size Modelling

The meanExcessPlot of the type 2 and type 3 claim sizes is given in figure 3.21 below:

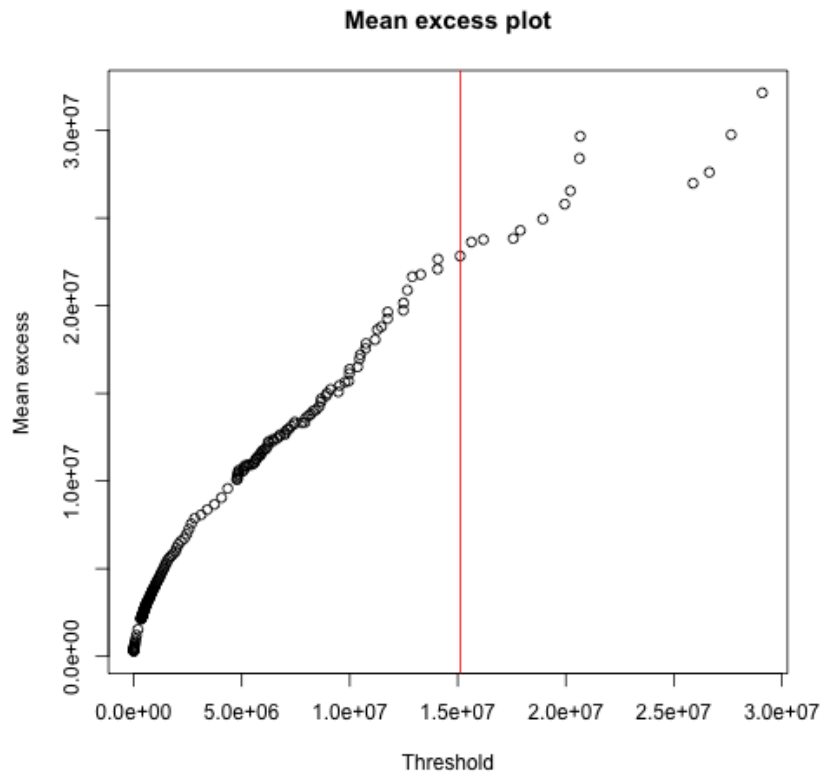


Figure 3.21: The mean excess plot for type 2 and 3 claim sizes

It is not obvious which threshold we should choose from the plot, but we try $b = 15\,108\,527$ for the claim sizes of type 2 and type 3 catastrophes. The over-threshold part data consists of 36 observations, which corresponds to an approximately upper 0.2% percentile of the data.

Central part:

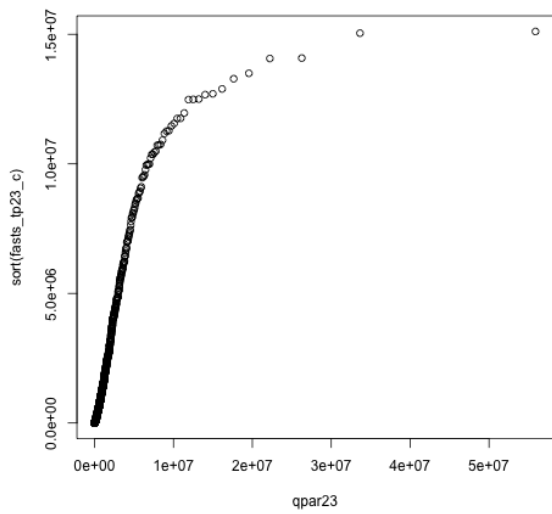
- Log-normal distribution:

The estimates of the mean and the standard deviation of $\log(Z)$ are given by:

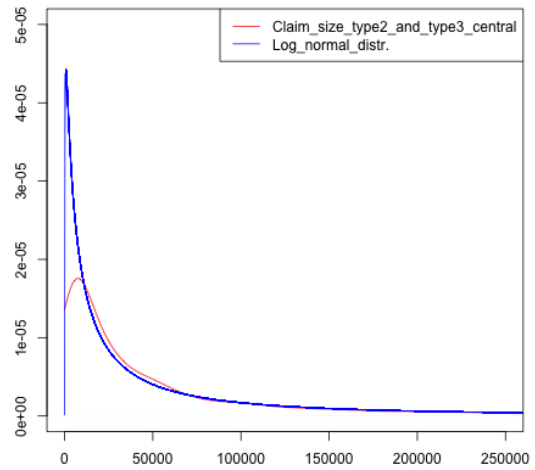
$$\bar{y}_{2\&3} = 10.26 \quad \text{and} \quad s_{y_{2\&3}} = 1.89.$$

Hence, the estimates of the parameters of Z are given by:

$$\hat{\xi}_{2\&3} = 1.72 \cdot 10^5 \quad \text{and} \quad \hat{\sigma}_{2\&3} = 1.89.$$



QQ-plot: Log-normal vs. Type 2 and 3 claim sizes



Density plot: Log-normal vs. Type 2 and 3 claim sizes

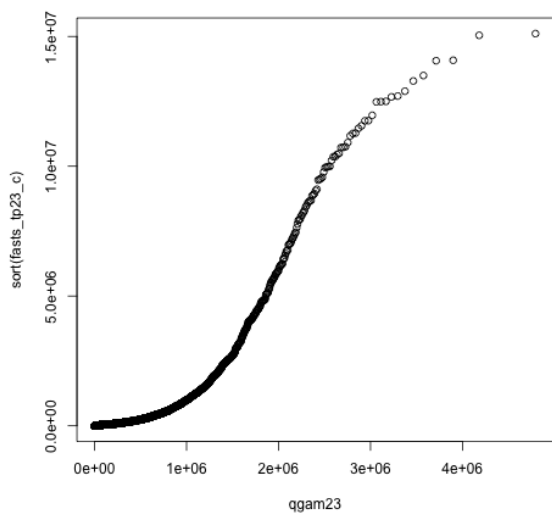
Figure 3.22: Fitting of the claim sizes of type 2 and type 3 by the log-normal distribution

According to figure 3.22 above, the log-normal distribution gives bad fit.

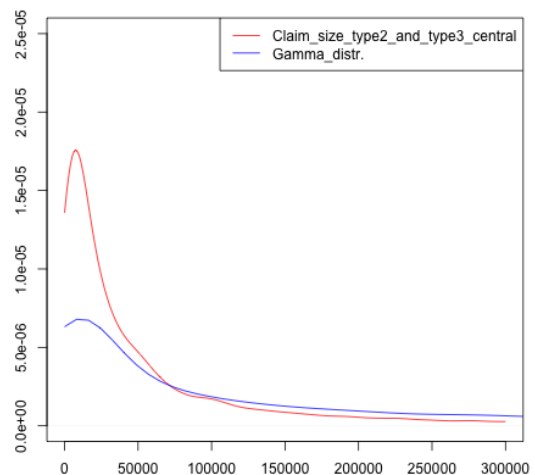
- Gamma distribution:

By the ML method, the estimates of the parameters are given by:

$$\hat{\xi}_{2\&3} = 2.06 \cdot 10^5 \quad \text{and} \quad \hat{\alpha}_{2\&3} = 0.34.$$



QQ-plot: Gamma vs. Type 2 and 3 claim sizes



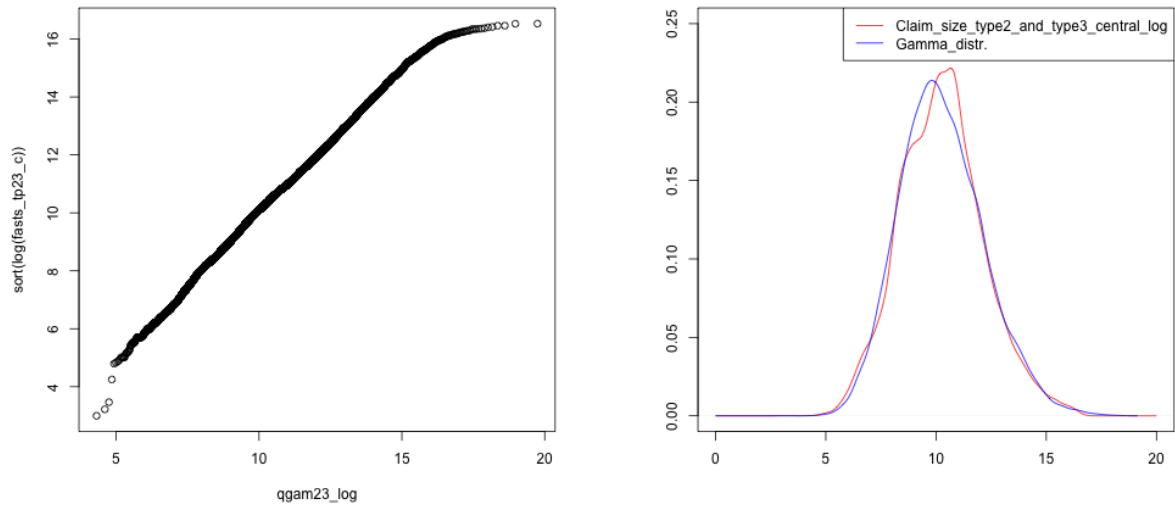
Density plot: Gamma vs. Type 2 and 3 claim sizes

Figure 3.23: Fitting of the claim sizes of type 2 and type 3 by the Gamma distribution

The Gamma distributions gives a bad fit as we can see in figure 3.23. Now, we try fitting Gamma

distribution on the logarithm of the data, the estimates of parameters are given by:

$$\hat{\xi}_{2\&3,\log} = 10.26 \quad \text{and} \quad \hat{\alpha}_{2\&3,\log} = 28.81. \quad (3.6)$$



Q-Q-plot: Gamma vs. Type 2 and 3 claim sizes on the log

Density plot: Gamma vs. Type 2 and 3 claim sizes on the log

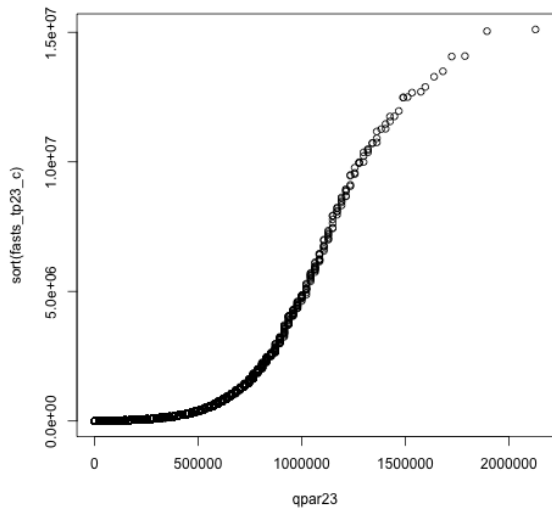
Figure 3.24: Fitting of the claim sizes of type 2 and type 3 on the log scale by the Gamma distribution

On the log scale, the Gamma distribution fits the claim sizes of type 2 and type 3 catastrophes very well as figure 3.24 shown above.

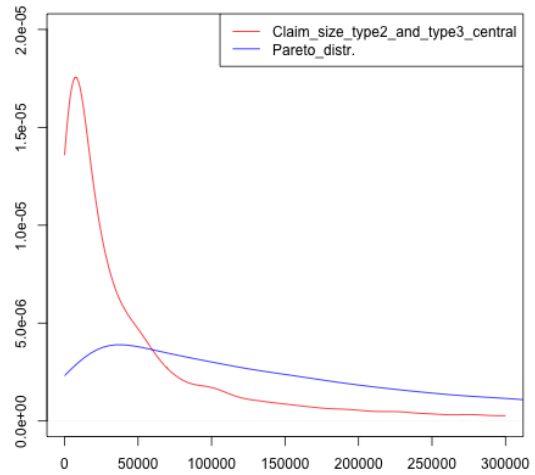
- Pareto distribution:

By the ML method, the estimates of the parameters are given by:

$$\hat{\beta}_{2\&3} = 9.59 \cdot 10^{19} \quad \text{and} \quad \hat{\alpha}_{\beta_{2\&3}} = 4.68 \cdot 10^{14}.$$



QQ-plot: Pareto vs. Type 2 and 3 claim sizes



Density plot: Pareto vs. Type 2 and 3 claim sizes

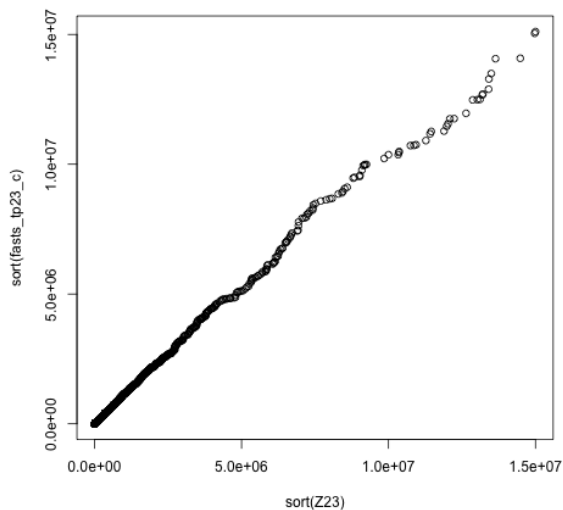
Figure 3.25: Fitting of the claim sizes of type 2 and type 3 by the Pareto distribution

According to figure 3.25, the Pareto distribution does not fit the claim sizes of type 2 and type 3 catastrophes well.

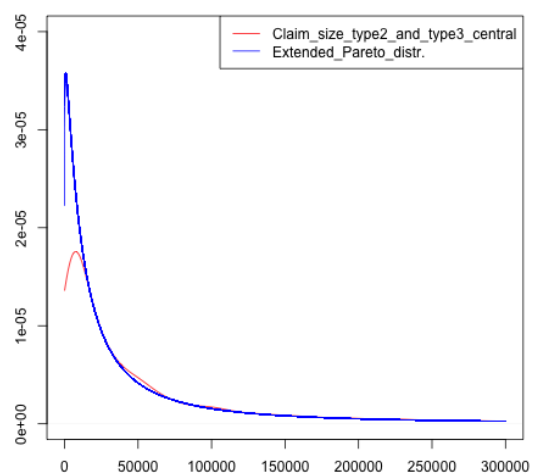
- Extended Pareto distribution:

The estimates of the parameters are given by:

$$\hat{\alpha}_{2\&3} = 0.80, \quad \hat{\beta}_{2\&3} = 1.70 \cdot 10^4 \quad \text{and} \quad \hat{\theta}_{2\&3} = 1.08.$$



QQ-plot: Extended Pareto vs. Type 2 and type 3 claim sizes



Density plot: Extended Pareto vs. Type 2 and type 3 claim sizes

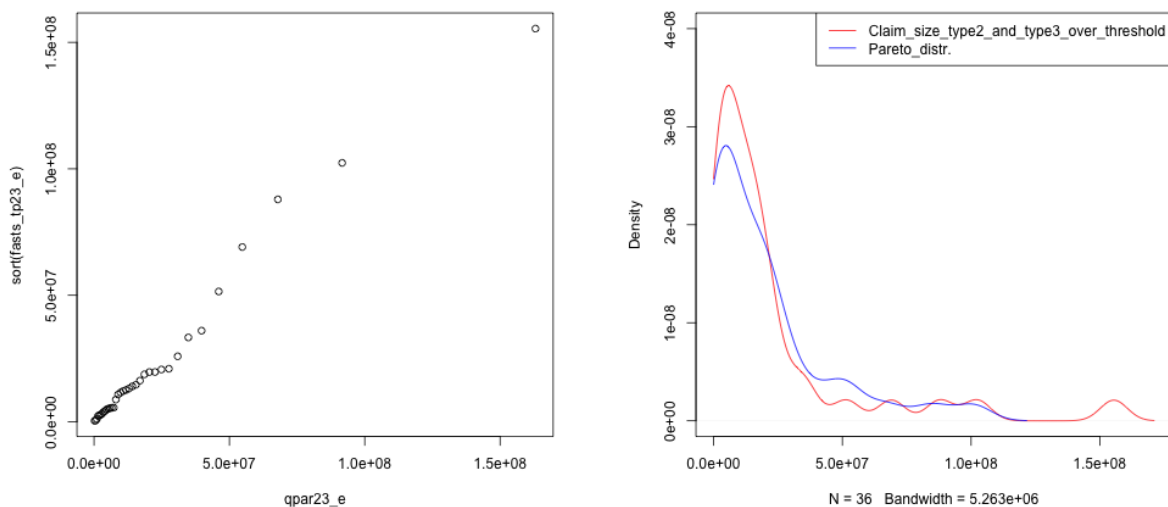
Figure 3.26: Fitting of the claim sizes of type 2 and type 3 by the extended Pareto distribution

According to figure 3.26, the extended Pareto gives a good fit to the data, but once more, we prefer the log-Gamma distribution.

Over-threshold part:

The over-threshold part is based on 36 data observations. By the ML method, the estimates of the Pareto parameters are given by:

$$\hat{\beta}_{2\&3,over} = 3.44 \cdot 10^7 \quad \text{and} \quad \hat{\alpha}_{\beta_{2\&3,over}} = 2.45.$$



QQ-plot: Pareto vs. Type 2 and 3 over-threshold claim sizes

Density plot: Pareto vs. Type 2 and 3 over-threshold claim sizes

Figure 3.27: Fitting of the over-threshold claim sizes of type 2 and type 3 by the Pareto distribution

As we can see in figure 3.27, the Pareto distribution seems to fit rather well, which indicates that the chosen threshold is adequate, but it is a little difficult to judge based on so few observations.

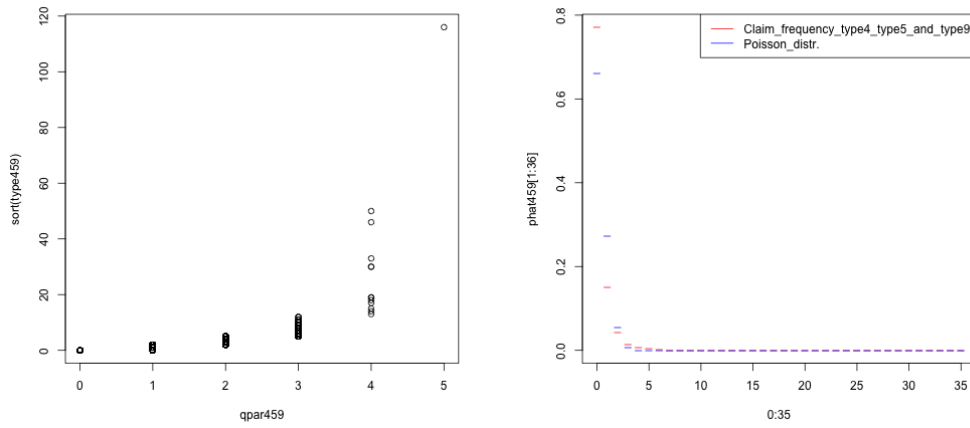
3.3.3 Catastrophe Type 4, Type 5 and Type 9

3.3.3.1 Claim Frequency Modelling

- Poisson distribution:

The estimate of the parameter $\mu_{4\&5\&9}$ is given by:

$$\hat{\mu}_{4\&5\&9} = \frac{\text{Claim numbers of type 4, 5 and 9}}{A} = \frac{6043}{431 \times 34} = 0.41.$$



QQ-plot: Poisson vs. Type 4, 5, 9 catastrophes

Density plot: Poisson vs. Type 4, 5, 9 catastrophes

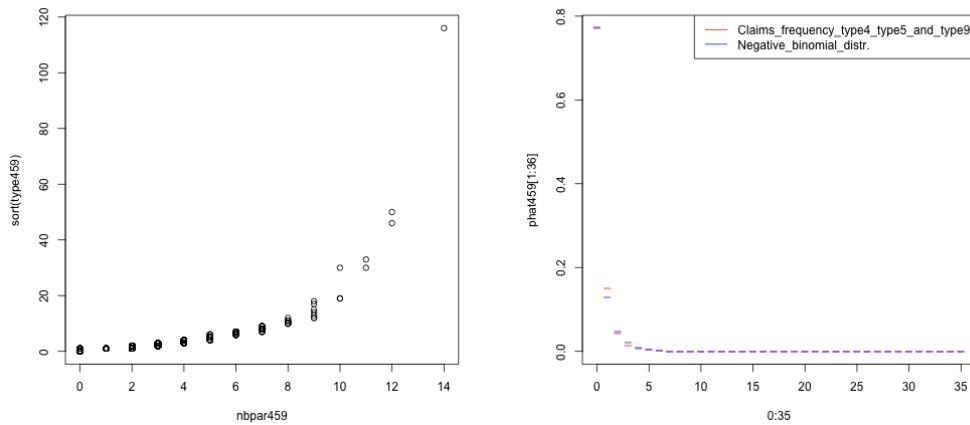
Figure 3.28: Fitting of the catastrophe frequency of type 4, 5 and 9 by the Poisson distribution

The QQ-plot in figure 3.28 indicates that the Poisson distribution does not give a good fit to the catastrophes of type 4, 5 and 9.

- Negative binomial distribution:

The estimates of the parameters $\xi_{4&5&9}$ and $\alpha_{4&5&9}$ are given by:

$$\hat{\xi}_{4&5&9} = 0.41 \quad \text{and} \quad \hat{\alpha}_{4&5&9} = 0.28.$$



QQ-plot: NB vs. Type 4, 5, 9 catastrophes

Density plot: NB vs. Type 4, 5, 9 catastrophes

Figure 3.29: Fitting of the catastrophe frequency of type 4, 5 and 9 by the negative binomial distribution

When fitting to the number of catastrophes of type 4, 5 and 9, the negative binomial distribution seems to give a slightly better fit than the Poisson as we can see in figure 3.29.

3.3.3.2 Claim Size Modelling

The meanExcessPlot of type 4, type 5 and type 9 claim sizes is given in figure 3.30 below:

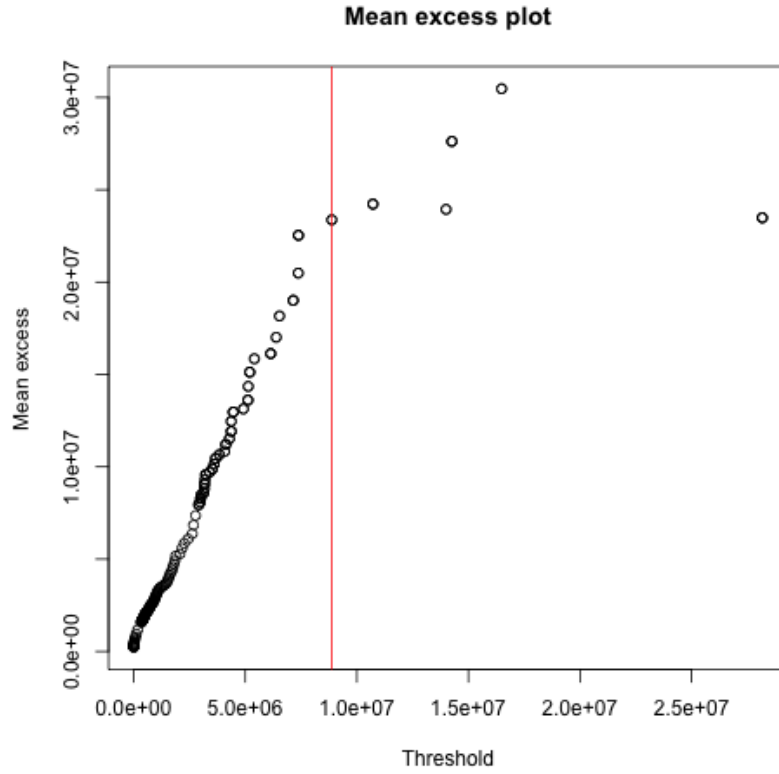


Figure 3.30: The mean excess plot for type 4, type 5 and type 9 claim sizes

Due to the small amount of data, it is impossible to choose the threshold based on the mean excess plot. Simply set b to the upper 0.2% quantile, i.e. $b = 8876982$, as for the remaining catastrophe types.

Central part:

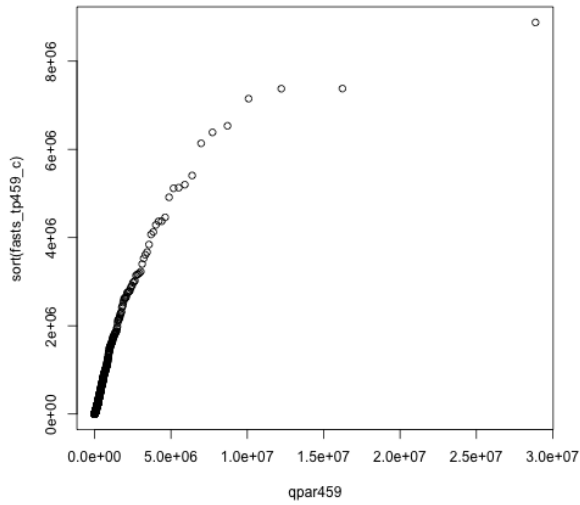
- Log-normal distribution:

The estimates of the mean and the standard deviation of $\log(Z)$ are given by:

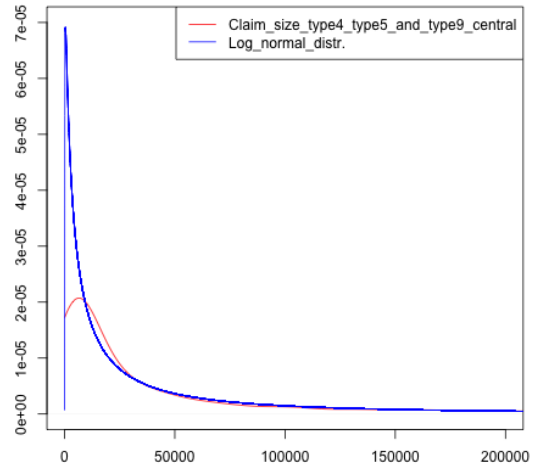
$$\bar{y}_{4\&5\&9} = 9.93 \quad \text{and} \quad s_{y_{4\&5\&9}} = 1.98.$$

Hence, the estimates of the parameters of Z are given by:

$$\hat{\xi}_{4\&5\&9} = 1.45 \cdot 10^5 \quad \text{and} \quad \hat{\sigma}_{4\&5\&9} = 1.98.$$



QQ-plot: Log-normal vs. Type 4, 5, 9 claim sizes



Density plot: Log-normal vs. Type 4, 5, 9 claim sizes

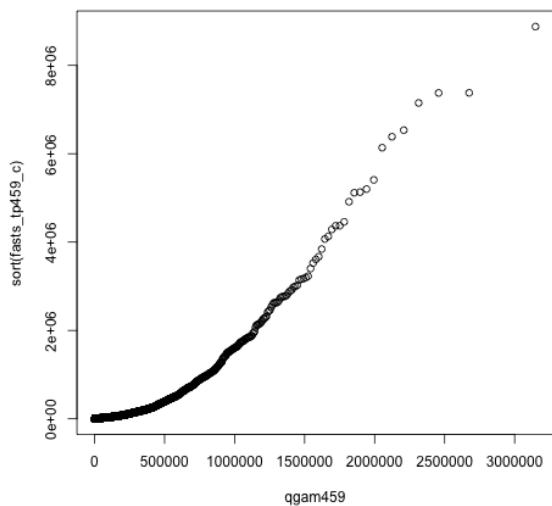
Figure 3.31: Fitting of the claim sizes of type 4, 5 and 9 by the log-normal distribution

It is clear to see in figure 3.31 that the log-normal distribution is not an ideal model for the claim sizes of catastrophe type 4, 5 and 9.

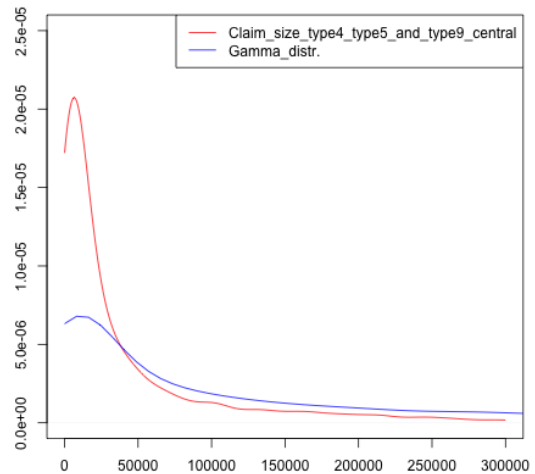
- Gamma distribution:

The estimates of the Gamma distribution are given by:

$$\hat{\xi}_{4\&5\&9} = 1.57 \cdot 10^5 \quad \text{and} \quad \hat{\alpha}_{4\&5\&9} = 0.33.$$



QQ-plot: Gamma vs. Type 4, 5, 9 claim sizes



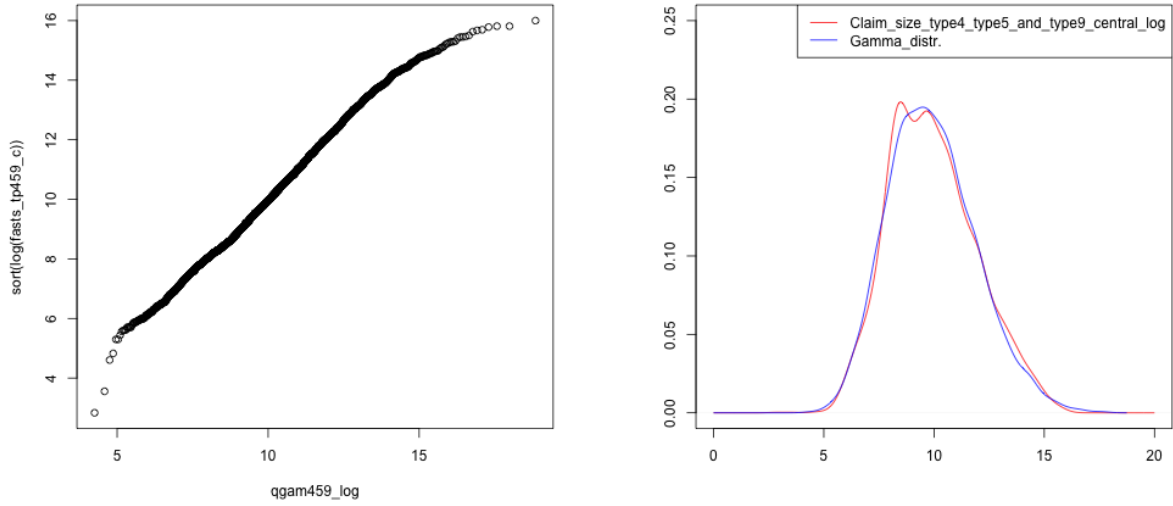
Density plot: Gamma vs. Type 4, 5, 9 claim sizes

Figure 3.32: Fitting of the claim sizes of type 4, 5 and 9 by the Gamma distribution

According to figure 3.32, the Gamma distribution does not fit the data well. Trying the Gamma on the

logarithm of the data, the estimates of the parameters are now given by:

$$\hat{\xi}_{4\&5\&9,\log} = 9.93 \quad \text{and} \quad \hat{\alpha}_{4\&5\&9,\log} = 25.24.$$



QQ-plot: Gamma vs. Type 4, 5, 9 claim sizes on the log

Density plot: Gamma vs. Type 4, 5, 9 claim sizes on the log

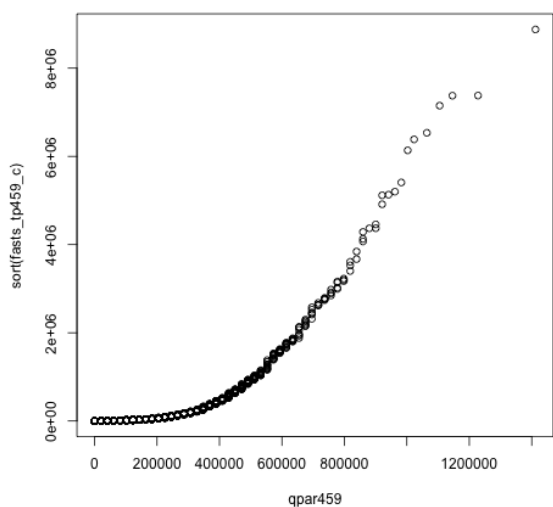
Figure 3.33: Fitting of the claim sizes of type 4, 5 and 9 on the log scale by the Gamma distribution

The log-Gamma distribution describes the claim sizes of catastrophe type 4, 5 and 9 very well as figure 3.33 shown.

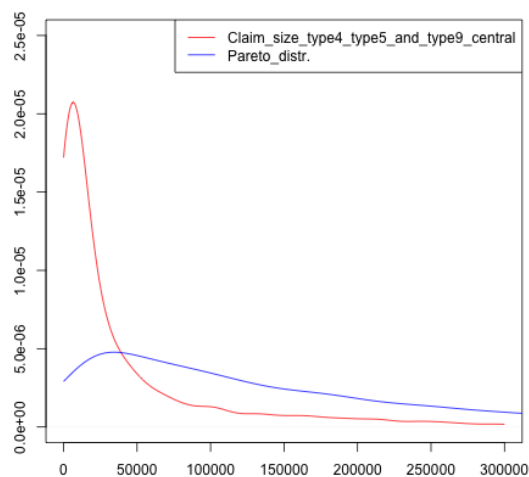
- Pareto distribution:

The estimates of the parameters are given by:

$$\hat{\beta}_{4\&5\&9} = 9.22 \cdot 10^{19} \quad \text{and} \quad \hat{\alpha}_{\beta_{4\&5\&9}} = 5.90 \cdot 10^{14}.$$



QQ-plot: Pareto vs. Type 4, 5, 9 claim sizes



Density plot: Pareto vs. Type 4, 5, 9 claim sizes

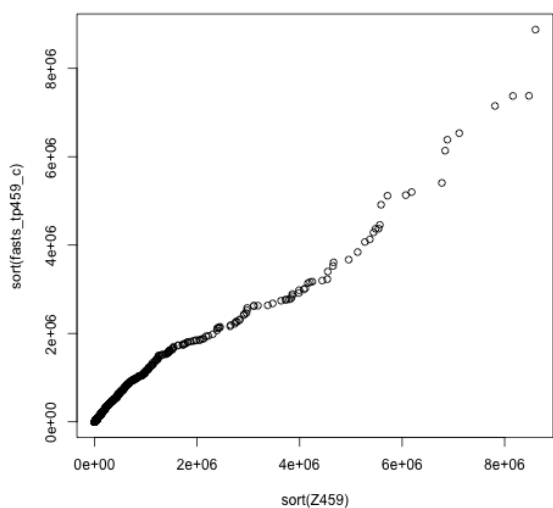
Figure 3.34: Fitting of the claim sizes of type 4, 5 and 9 by the Pareto distribution

Observe from figure 3.34 above, the Pareto distribution gives a bad fit to the central part of the data.

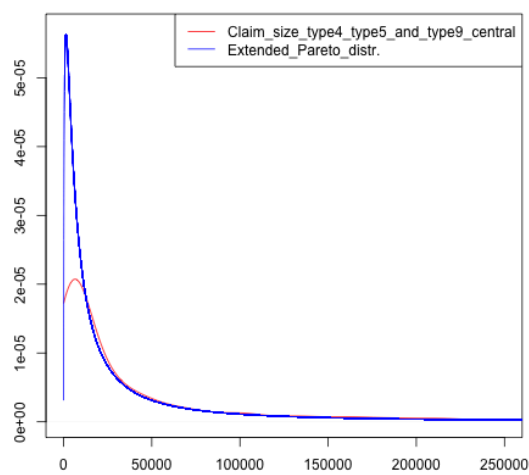
- Extended Pareto distribution:

The estimated parameters are given by:

$$\hat{\alpha}_{4\&5\&9} = 0.63, \quad \hat{\beta}_{4\&5\&9} = 4.78 \cdot 10^3 \quad \text{and} \quad \hat{\theta}_{4\&5\&9} = 1.47.$$



QQ-plot: Extended Pareto vs. Type 4, 5, 9 claim sizes



Density plot: Extended Pareto vs. Type 4, 5, 9 claim sizes

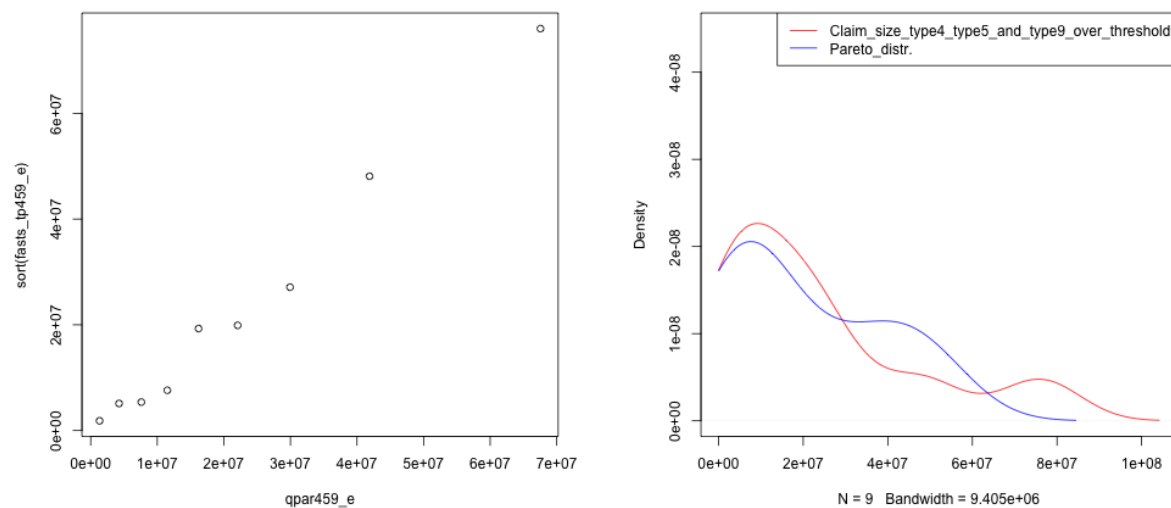
Figure 3.35: Fitting of the claim sizes of type 4, 5 and 9 by the extended Pareto distribution

According to figure 3.35, the extended Pareto gives a rather good fit to the data. Still, we choose the log-Gamma.

Over-threshold part:

The over-threshold part data consists of 9 observations. By the ML method, the estimates of the Pareto parameters are given by:

$$\hat{\beta}_{4\&5\&9,over} = 1 \cdot 10^{11} \quad \text{and} \quad \hat{\alpha}_{\beta_{4\&5\&9,over}} = 4.28 \cdot 10^4.$$



QQ-plot: Pareto vs. Type 4, 5, 9 over-threshold claim sizes

Density plot: Pareto vs. Type 4, 5, 9 over-threshold claim sizes

Figure 3.36: Fitting of the over-threshold claim sizes of type 4, 5 and 9 by the Pareto distribution

As we can see in figure 3.36, due to the very few observations, it is difficult to judge the fit, but it does not seem to be unreasonable.

3.4 Summary

After fitting different distributions to the number of catastrophes and the amounts of compensation, a short summary is given as follows:

For the number of catastrophes, both in the common model and the separate models, the negative binomial distribution always fits the data better than the Poisson. This is because that the assumed μ_j in the negative binomial model is allowed to vary from municipality to municipality. An exception is for the catastrophes of type 4&5&9 where the difference between the Poisson and the negative binomial distribution is much smaller.

For the amounts of compensation, in the central part, the log-Gamma distribution gives excellent fit both in the common model and the separate models. As mentioned in section 2.3.2, the empirical distribution could be an alternative, as far as the number of observations is large. However, there is usually larger uncertainty in estimates based on the empirical distribution than in estimates based on the parametric models. Since we are able to find a parametric distribution that fits the claim sizes very

well here, the empirical distribution will not be considered in our case. In the extreme right tail, the Pareto distribution fits the data quite well, especially in the common model and the model for the data of catastrophe type 1. In the model for the claim sizes of catastrophe type 2&3 and type 4&5&9, since the number of observations in the extreme right tail is very few based on the thresholds we have chosen, the Pareto does not fit as well. However, as explained in section 2.3.3.1, it is important to take into account both the bias and the variance in the over-threshold part. By setting a higher threshold, the Pareto distribution could fit better, but then the variance would be much larger. Contrarily, with a lower threshold, the Pareto distribution would not fit the data well although there would be smaller variance. We have been trying to reach a compromise between the bias and the variance, therefore we have been left with the results as shown above.

Since it is the same distributions that fit the common model and the separate models well, we could probably suggest the common model over the separate models. In order to judge whether the common model can be used instead of the separate models, it is necessary to check if the parameters are significantly different between the three separate models and the common model. If there is significant difference, then the common model can not be used. If not, the common model is preferable due to smaller uncertainty. The parameter difference can be checked by means of computing a confidence interval based on the bootstrap estimates obtained using algorithm 2 (see section 4.2 for detailed explanation of bootstrap confidence interval). Let the confidence level be 95%. The bootstrap estimates from the three separate models for \mathcal{N} are aggregated before constructing the confidence intervals. The parameter confidence intervals of the models for \mathcal{N} are given as follows:

		Common model	Compounded model
<i>Negative binomial</i>	ξ	(5.69, 5.90)	(5.71, 5.87)
	α	(1.04, 1.10)	(1.77, 1.86)

Table 3.1: Comparison of the parameter confidence intervals of the model for \mathcal{N}

From the table above we see that the confidence interval for ξ based on the common model almost overlaps the one based on the compounded model. However, there is a larger difference in the confidence interval for α from the common model and the compounded model, indicating that there is a difference between the common and the separate models for claim numbers.

The parameter confidence intervals of the models for $\mathbf{Z}_{\leq b}$ and $\mathbf{Z}_{> b}$ are given as follows:

		Common model	Type 1	Type 2&3	Type 4&5&9
<i>log-Gamma</i>	ξ	(9.94, 9.97)	(9.84, 9.87)	(10.23, 10.29)	(9.87, 10.00)
	α	(30.95, 31.60)	(33.14, 33.95)	(28.21, 29.49)	(24.19, 26.42)
<i>Pareto</i>	α	(1.45, 4.69)	(2.53, $9.32 \cdot 10^4$)	(1.12, $6.70 \cdot 10^4$)	(1.49, $1.17 \cdot 10^5$)
	β	($1.07 \cdot 10^7$, $5.43 \cdot 10^7$)	($1.74 \cdot 10^7$, $1 \cdot 10^{11}$)	($9.49 \cdot 10^6$, $1 \cdot 10^{11}$)	($1.34 \cdot 10^7$, $1 \cdot 10^{11}$)

Table 3.2: Comparison of parameter confidence intervals of the models for \mathbf{Z}

In the model for $\mathbf{Z}_{\leq b}$, the differences between the log-Gamma parameters of the common model and the separate models are quite large since the confidence intervals do not overlap. In the extreme part, the

confidence intervals do overlap, indicating that there is no significant difference between the parameters from the common model and the separate models, although the confidence intervals are quite wide because of the few observations we have in the extreme right tail.

The parameters in the negative binomial distribution and the log-Gamma distribution are significantly different. This indicates that claim numbers and sizes from different catastrophe types behave differently, and the common model is not suggested. The choice of the model will be discussed more in the conclusion chapter associated with the estimated reserve and its uncertainty.

Chapter 4

Total Loss and Reserve

In this chapter, the model for \mathcal{N} and \mathbf{Z} will be put together to construct the model for the total loss \mathcal{X} . The best fitting distributions from the previous chapter, i.e. the negative binomial and the log-Gamma along with the Pareto distribution are applied to the data. The reserve q_ϵ will be estimated from the loss model by means of the Monte Carlo simulations. Possible parameter errors from the underlying models will be converted into the uncertainty assessment of the reserve using the nested bootstrap Monte Carlo algorithm. Thereafter, the estimated reserve based on the common model will be compared with the one based on a combination of the separate models. At the end of this chapter, a brief summary will be given.

4.1 Total Loss and Reserve Simulation

Recall the model for the total loss based on all catastrophes from (2.62) and the loss model aggregated by separate catastrophe types from (2.63). The negative binomial distribution is chosen for the number of catastrophes \mathcal{N} , and the log-Gamma distribution is chosen for the central part of the amounts of compensation $\mathbf{Z}_{\leq b}$ and the Pareto distribution is supposed to be the default model for the extreme part $\mathbf{Z}_{> b}$. The models depend on the estimated parameters $(\hat{\xi}_{\text{NB}}, \hat{\alpha}_{\text{NB}})$, $(\hat{\xi}_{\text{Gamma,log}}, \hat{\alpha}_{\text{Gamma,log}})$ and $(\hat{\alpha}_{\text{Pareto}}, \hat{\beta}_{\text{Pareto}})$, respectively. Therefore, the model for the total loss is constructed by applying line 1 to line 10 in algorithm 3 where line 3 and line 7 are modified for the negative binomial and the log-Gamma distribution, respectively. The reserve is simulated by Monte Carlo using algorithm 3, line 3 and line 7 have been modified accordingly. The number of Monte Carlo simulations is chosen to be $m = 10000$ and $\epsilon = 0.01$. The uncertainty in the reserve estimates is assessed by the nested bootstrap explained in algorithm 4, line 2, 6, 9, 10 and 14 have been modified accordingly. The number of bootstrap replications is chosen to be $m_b = 1000$. Hence there is a total $m \times m_b$ simulations in the nested bootstrap procedure.

4.2 Reserve Comparison

Denote the estimates of the reserve based on common model as $\hat{q}_{\epsilon, \text{Total}}^*$ and the one based on separate models as $\hat{q}_{\epsilon, \text{Separate}}^*$. The estimates of the reserve by algorithm 3 are given by:

$$\begin{aligned}\hat{q}_{\epsilon, \text{Total}}^* &= 1.20 \cdot 10^9, \\ \hat{q}_{\epsilon, \text{Separate}}^* &= 1.36 \cdot 10^9.\end{aligned}\tag{4.1}$$

One way of measuring the uncertainty in the reserve is to compute a confidence interval. When constructing a confidence interval, a confidence level is first selected. An often-used confidence level is 95%, which means that there is a 95% chance that the interval would include the true value. The higher the confidence level is, the more trustworthy it is that the true value lies within the interval [Devore and Berk, 2007, p.375].

The standard deviation is a measure of how spread the data is, and is another measure of uncertainty in the reserve. A low standard deviation value indicates that the data are close to the mean value of the data set. The sample standard deviation $\text{sd}_{\text{boot}}(\hat{q}_\epsilon^{**})$ of the m_b bootstrap estimates of the reserve is given by (2.57).

The confidence interval and the standard deviation of \hat{q}_ϵ^{**} can be calculated from the results in the nested bootstrap algorithm. Line 1 to 17 in algorithm 4 will be first run for the common model and the separate models, respectively. Before estimating the reserve for the combined model, the estimates of the total loss from separate models have to be aggregated. Hence we have m_b sets of m estimates for the total loss $\mathcal{X}_{\text{Total}}$ from the common model and m_b sets of m estimates for the total loss $\mathcal{X}_{\text{Separate}}$ summed up by the separate models. The bootstrap estimate of the reserve $\hat{q}_{\epsilon, \text{Total}}^{**}$ and $\hat{q}_{\epsilon, \text{Separate}}^{**}$ are derived from the upper ϵ -percentile of the corresponding \mathcal{X} .

The distribution of $\hat{q}_{\epsilon, \text{Total}}^{**}$ and $\hat{q}_{\epsilon, \text{Separate}}^{**}$ can be illustrated by a histogram plot in R.

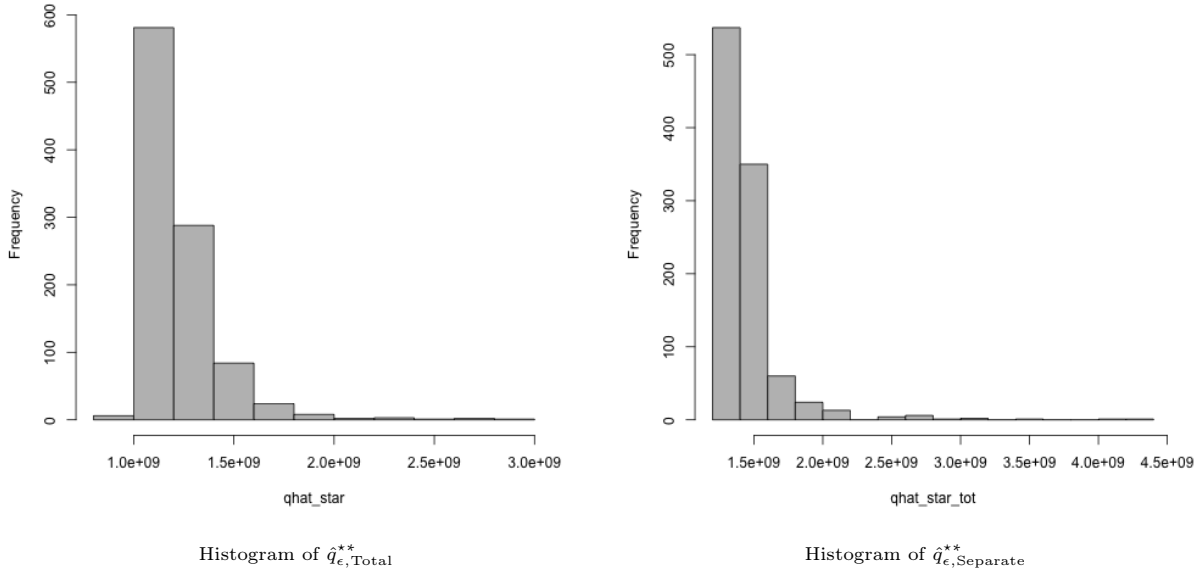


Figure 4.1: Illustration of the distribution of $\hat{q}_{\epsilon, \text{Total}}^{**}$ and $\hat{q}_{\epsilon, \text{Separate}}^{**}$

Neither $\hat{q}_{\epsilon, \text{Total}}^{**}$ nor $\hat{q}_{\epsilon, \text{Separate}}^{**}$ seems to be normally distributed. Therefore the confidence interval based on a normal distribution cannot be used. Instead, a more general confidence interval based on bootstrap percentiles can be constructed. An explanation of the bootstrap confidence interval is as follows: Let G be the cumulative distribution of \hat{q}_ϵ^{**} . Then $G^{-1}(\alpha) = \hat{q}_{\epsilon, (\alpha)}^{**}$, which is the $(100 \cdot \alpha)^{\text{th}}$ percentile of the bootstrap distribution. Then the $(1 - \alpha)\%$ percentile interval of \hat{q}_ϵ^{**} can be written as:

$$\begin{aligned} (\hat{q}_{\epsilon, L}^{**}, \hat{q}_{\epsilon, U}^{**}) &= [G^{-1}(\alpha/2), G^{-1}(1 - \alpha/2)] \\ &= [\hat{q}_{\epsilon, (\alpha/2)}^{**}, \hat{q}_{\epsilon, (1-\alpha/2)}^{**}]. \end{aligned} \tag{4.2}$$

The percentile interval presented in (4.2) is an ideal situation when the number of bootstrap replications m_b goes to infinity. In practice, when m_b is finite, we sort all the bootstrap estimates in ascending order: $\hat{q}_{\epsilon,(1)}^{**} \leq \hat{q}_{\epsilon,(2)}^{**} \leq \dots \leq \hat{q}_{\epsilon,(m_b)}^{**}$. Then the $(100 \cdot \alpha)^{th}$ empirical percentile of the \hat{q}_{ϵ}^{**} 's is just the $(m_b \cdot \alpha)^{th}$ value in the ordered \hat{q}_{ϵ}^{**} 's [Efron and Tibshirani, 1994, p.168–171]. Hence, the approximate $(1 - \alpha)100\%$ percentile interval of \hat{q}_{ϵ}^{**} can be written as:

$$\left(\hat{q}_{\epsilon,(\frac{\alpha}{2} \cdot m_b)}^{**}, \hat{q}_{\epsilon,[(1-\frac{\alpha}{2}) \cdot m_b]}^{**} \right). \quad (4.3)$$

By (4.3), with 95% confidence level, i.e. $\alpha = 0.05$ and the number of the bootstrap replications $m_b = 1000$, the bootstrap confidence interval for $\hat{q}_{\epsilon,Total}^{**}$ based on the common model is given by:

$$CI_{\hat{q}_{\epsilon,Total}^{**}} = (1.03 \cdot 10^9, 1.68 \cdot 10^9).$$

The standard deviation of the bootstrap estimates $\hat{q}_{\epsilon,Total}^{**}$ is given by:

$$sd_{boot}(\hat{q}_{\epsilon,Total}^{**}) = 1.95 \cdot 10^8.$$

Similarly, the 95% confidence interval for $\hat{q}_{\epsilon,Separate}^{**}$ based on the separate models is given by:

$$CI_{\hat{q}_{\epsilon,Separate}^{**}} = (1.26 \cdot 10^9, 2.04 \cdot 10^9).$$

The standard deviation of the bootstrap estimates $\hat{q}_{\epsilon,Separate}^{**}$ is given by:

$$sd_{boot}(\hat{q}_{\epsilon,Separate}^{**}) = 2.48 \cdot 10^8.$$

4.3 Summary

After comparing the bootstrap confidence intervals and the standard deviations a short summery is given as follows:

The estimate of the reserve based on the common model is smaller than the estimate of the one based on the separate models. The confidence interval of $\hat{q}_{\epsilon,Total}^{**}$ is narrower than the confidence interval of $\hat{q}_{\epsilon,Separate}^{**}$, indicating that the uncertainty in the reserve estimate from the common model is smaller than the one from the separate models. The standard deviation of $\hat{q}_{\epsilon,Total}^{**}$ is also smaller than the standard deviation of $\hat{q}_{\epsilon,Separate}^{**}$. The results are not something surprising, since there are more estimated parameters involved in the separate models, and there are more data in the common model for estimation.

Chapter 5

Conclusion

The total loss model \mathcal{X} consists of two components, the occurrences of catastrophes \mathcal{N} and the amounts of compensation \mathbf{Z} . The amounts of compensation \mathbf{Z} are separated by a threshold b , where the over-threshold part $\mathbf{Z}_{>b}$ is supposed to follow the Pareto distribution according to the Pickand's theorem, and the central part $\mathbf{Z}_{\leq b}$ is supposed to follow some parametric distribution. Different distributions are tested on \mathcal{N} and $\mathbf{Z}_{\leq b}$. Both in the common model and the separate models, the negative binomial distribution and the log-Gamma distribution give best fit to \mathcal{N} and $\mathbf{Z}_{\leq b}$, respectively. For $\mathbf{Z}_{>b}$, the Pareto distribution describes the data rather well where a balance between the model bias and the variance has been taken into account. The reserve q_ϵ , which is the upper ϵ -percentile of the total loss \mathcal{X} , is estimated through the Monte Carlo simulation. Possible sources of uncertainty have been discussed and evaluated with a combination of the bootstrap Monte Carlo scheme.

The reserve estimate from a combination of the separate models is larger than the one from the common model. In general insurance, when it comes to the reserve estimation, it is very important not to underestimate the reserve. It is always better to risk overestimating rather than to risk underestimating the reserve since it would not be as a big crisis for insurance companies to have some extra fund (not too much either) as a shortage of fund. For the estimation uncertainty, according to the confidence intervals and the standard deviations computed in the previous chapter, the reserve estimate based on the separate models has a larger uncertainty than the one from the common model, which is not unexpected. Thus, the choice of the model is based on the following arguments:

- The confidence intervals of the distribution parameters given in section 3.4 indicate a significant difference of the parameters in the separate models and the common model.
- Although the uncertainty in the reserve estimate based on the separate models is larger than the one based on the common model, it is more "correct" to have a higher reserve estimate to ensure that the Pool is able to cover significant losses under worst-case situations.
- The six catastrophe types have already been merged into three larger groups. In this way, the uncertainty in the underlying models has already been reduced under modelling.

In conclusion, on the premise that our models are reasonable and correct, the risk of the Pool should be evaluated by a break-down into catastrophe type model. However, we would suggest merging the similar catastrophe types for reduced uncertainty.

Reference

- [NNP, 2016] (2016). Organization of the Norwegian Natural Perils Pool. <http://www.naturskade.no/en/Main/Home/Organisation/>. Accessed:2016-05-12.
- [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*. Duxbury Pacific Grove, CA.
- [de Seguros, 2008] de Seguros, C. d. C. (2008). Natural Catastrophes Insurance Cover. A Diversity of Systems. http://www.wfcatprogrammes.com/c/document_library/get_file?folderId=13442&name=DLFE-553.pdf. Accessed:2016-05-11.
- [Devore and Berk, 2007] Devore, J. L. and Berk, K. N. (2007). *Modern Mathematical Statistics with Applications*. Cengage Learning.
- [Duong, 2001] Duong, T. (2001). An introduction to kernel density estimation. <http://www.mvstat.net/tduong/research/seminars/seminar-2001-05/>. Accessed: 2016-03-23.
- [Efron and Tibshirani, 1994] Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC press.
- [Eide, 2015] Eide, G. (2015). The Norwegian natural catastrophe compensation system in the past and the future. <http://www.conorsegurosdigital.com/almacen/pdf/the-norwegian-natural-catastrophe-compensation-system-in-the-past-and-the-future.pdf>. Accessed:2016-05-12.
- [Ginos, 2009] Ginos, B. F. (2009). Parameter Estimation for the Lognormal Distribution. <http://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=2927&context=etd>.
- [Hansen, 2009] Hansen, B. E. (2009). Lecture Notes on Nonparametrics. <http://www.ssc.wisc.edu/~bhansen/718/NonParametrics1.pdf>. Accessed: 2016-03-23.
- [ii Association, 2016] ii Association, S. (2016). The solvency capital requirement. <http://www.solvency-capital-requirement.com>. Accessed: 2016-04-05.
- [Lehmann, 1999] Lehmann, E. L. (1999). *Elements of Large-Sample Theory*. Springer Science & Business Media.

- [Mahadevan, 1997] Mahadevan, S. (1997). Monte Carlo Simulation. *MECHANICAL ENGINEERING-NEW YORK AND BASEL-MARCEL DEKKER*-, pages 123–146.
- [Mooney, 1997] Mooney, C. Z. (1997). *Monte Carlo Simulation*, volume 116. Sage Publications.
- [Ross, 2010] Ross, S. M. (2010). *Introduction to Probability Models*. Academic press.
- [Rubinstein and Kroese, 2008] Rubinstein, R. Y. and Kroese, D. P. (2008). *Simulation and the Monte Carlo Method*. John Wiley & Sons.
- [Scarrott and MacDonald, 2012] Scarrott, C. and MacDonald, A. (2012). A Review of Extreme Value Threshold Estimation and Uncertainty Qualification. <https://www.ine.pt/revstat/pdf/rs120102.pdf>. Accessed: 2016-04-09.
- [Sigman, 2010] Sigman, K. (2010). Inverse Transform Method. <http://www.columbia.edu/~ks20/4404-Sigman/4404-Notes-ITM.pdf>. Accessed: 2016-03-29.
- [Thomopoulos, 2012] Thomopoulos, N. T. (2012). *Essentials of Monte Carlo Simulation: Statistical methods for building simulation models*. Springer Science & Business Media.

Appendix A

Mathematical Arguments

A.1 The Double Rules of Expectation and Variance

The double rules of expectation and variance are useful tools when finding expectation and standard deviation of a conditional distribution. The conditional distribution is defined as follows: assume a random variable Y given $X = x$ when X and Y are continuous, the conditional probability density for Y is then given by:

$$f_{Y|X}(y|x) = \frac{f(x, y)}{f_X(x)}.$$

Hence,

$$f(x, y) = f_X(x)f_{Y|X}(y|x), \tag{A.1}$$

which is the joint density function of (x, y) . The mean and the variance of a conditional distribution depends on the variable that one conditions on. The conditional mean and variance of Y given that $X = x$ can be written as:

$$\xi(x) = E(Y|X = x) \quad \text{and} \quad \sigma^2(x) = \text{Var}(Y|X = x),$$

By (A.1) the expected value of a random variable Y can be written as:

$$E(Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} yf(x, y)dydx = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} yf(y|x)dy \right) f(x)dx = \int_{-\infty}^{\infty} \xi(x)f(x)dx.$$

We observe that the latter integral is just the $E[\xi(x)]$ by definition. Therefore, we have that:

$$E(Y) = E[\xi(x)]. \tag{A.2}$$

For the variance, by the general definition of variance, $\text{Var}(Y) = E(Y^2) - [E(Y)]^2$. Conditioning on X on both sides yields that:

$$\text{Var}(Y|X) = E(Y^2|X) - [E(Y|X)]^2. \tag{A.3}$$

By taking the expectation on both sides yields that:

$$E[\text{Var}(Y|X)] = E[E(Y^2|X)] - E\{[E(Y|X)]^2\} \tag{A.4}$$

By the law of iterated expectation, $E(Y) = E(E(Y|X))$. Hence, (A.4) can be written as:

$$E[\text{Var}(Y|X)] = E(Y^2) - E\{[E(Y|X)]^2\} \tag{A.5}$$

By taking the variance on $E(Y|X)$, we have:

$$\begin{aligned}\text{Var}(E(Y|X)) &= E\{[E(Y|X)]^2\} - \{E[E(Y|X)]\}^2 \\ &= E\{[E(Y|X)]^2\} - [E(Y)]^2.\end{aligned}\tag{A.6}$$

Add (A.5) and (A.6) together, the $E\{[E(Y|X)]^2\}$ -term is vanished, hence:

$$E(Y^2) - [E(Y)]^2 = E[\text{Var}(Y|X)] + \text{Var}(E(Y|X)),\tag{A.7}$$

which the left side of the equation is just $\text{Var}(Y)$. Therefore, the variance of Y is now given by:

$$\text{Var}(Y) = E[\sigma^2(X)] + \text{Var}[\xi(X)].\tag{A.8}$$

The rules of double expectation and variance can be proved to hold for discrete variables as well, and they have been used several places in the thesis.

A.2 The Mean and Standard Deviation of the Negative Binomial Distribution

Recall from (2.9) that:

$$E(N|\mu) = \text{Var}(N|\mu) = \lambda = \mu T.$$

By the conditional expectation and variance in (A.2) and (A.8) combined with (2.16) :

$$E(N) = E[E(N|\mu)] = E(\mu T) = E(\mu)T = \xi T$$

and

$$\begin{aligned}\text{sd}(N) &= \sqrt{E[\text{Var}(N|\mu)] + \text{Var}[E(N|\mu)]} \\ &= \sqrt{E(\mu T) + \text{Var}(\mu T)} \\ &= \sqrt{\xi T + \frac{\xi^2}{\alpha} T^2} \\ &= \sqrt{\xi T(1 + \xi T/\alpha)}.\end{aligned}$$

A.3 Inverse Transform Method

The motivation of the inverse transform method is that we want to simulate a random variable X from its cumulative distribution function F , i.e. to generate X such that:

$$\Pr(X \leq x) = F(x), \quad x \in \mathbb{R}.\tag{A.9}$$

Let F be a strictly increasing and continuous distribution function. Define $X = F^{-1}(U)$ (which is the inverse of the equation $F(X) = U$), where U is uniform distributed on the interval $[0, 1]$. Since F is increasing, we have that:

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

Hence:

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

Using the fact that $0 \leq F(x) \leq 1$ and the CDF of a uniform distribution $F_U(x) = x$, we have:

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

Thus, we have a simple general sampling technique using the uniform distribution [Sigman, 2010].

The cumulative distribution function of the Pareto distribution in section 2.3.1.3 is given by:

$$F(x) = 1 - (1 + x/\beta)^{-\alpha}, \quad x > 0. \quad (\text{A.10})$$

Set $F(x) = u$ and solve for x , we get the inverse CDF:

$$F^{-1}(u) = \beta\{(1 - u)^{-1/\alpha} - 1\}, \quad (\text{A.11})$$

which is used for sampling the Pareto random variables.

A.4 A Representation of the Extended Pareto

For showing the representation in (2.44), a little transformation is needed here:

Let $Z = X/Y$ where X and Y are independent and positive random variables with density functions $g_1(x)$ and $g_2(y)$, respectively. We have then:

$$\begin{aligned} F(z) &= \Pr(Z \leq z) = \Pr\left(\frac{X}{Y} \leq z\right) = \Pr(X \leq zY) \\ &= \int_0^\infty \Pr(X \leq zy)g_2(y)dy. \end{aligned}$$

By differentiating with respect to z , we get the density function of Z :

$$\begin{aligned} f(z) &= F'(z) = \frac{\partial}{\partial z} \int_0^\infty \Pr(X \leq zy)g_2(y)dy \\ &= \int_0^\infty \left(\frac{\partial}{\partial z} \Pr(X \leq zy)\right)g_2(y)dy, \end{aligned} \quad (\text{A.12})$$

where

$$\frac{\partial}{\partial z} (\Pr(X \leq zy)) = \frac{\partial}{\partial z} F_X(zy) = F'_X(zy) \cdot y = g_1(zy) \cdot y.$$

Hence, (A.12) can be now expressed by:

$$f(z) = \int_0^\infty g_1(zy) \cdot y \cdot g_2(y)dy. \quad (\text{A.13})$$

Let $X = \theta G_1$ and $Y = \alpha G_2$, where G_1 and G_2 are Gamma distributed with mean 1 and shape θ and α , respectively. Then $g_1(x) = x^{\theta-1}e^{-x}/\Gamma(\theta)$ and $g_2(y) = y^{\alpha-1}e^{-y}/\Gamma(\alpha)$. By substituting $x = y(1+z)$ in the integral:

$$\begin{aligned} f(z) &= \frac{z^{\theta-1}}{\Gamma(\theta)\Gamma(\alpha)} \int_0^\infty y^{\alpha+\theta-1} e^{-y(1+z)} dy \\ &= \frac{z^{\theta-1}}{\Gamma(\theta)\Gamma(\alpha)} \frac{1}{(1+z)^{\alpha+\theta}} \int_0^\infty x^{\alpha+\theta-1} e^{-x} dx. \end{aligned}$$

Using the fact that $\Gamma(s) = \int_0^\infty t^{s-1}e^{-t}dt$, we obtain:

$$f(z) = \frac{\Gamma(\alpha + \theta)}{\Gamma(\theta)\Gamma(\alpha)} \frac{z^{\theta-1}}{(1+z)^{\alpha+\theta}},$$

[Bølviken, 2014, p.335], which is on the form of the extended Pareto distribution.

A.5 Kernel Density Estimation

For estimating the probability density non-parametrically, a histogram is the simplest way to go. The histogram is constructed by dividing the intervals of the data into sub-intervals, which also known as "bins". Each time when the data falls into a particular sub-interval, a block of bandwidth of the sub-interval is then placed on top of it. The main concern here is that the smoothness of the histogram depends on the bandwidth. With a smaller bandwidth gives a smoother histogram than the one with large bandwidth. Another problem is the choice of end points of the bins. Different choice of end points of bins could yield a total different estimate for the density. Therefore the choice of both the bandwidth and the end point can be quite tricky. The kernel density estimators are an improvement over the histogram since it is independent of the end points of the bins, but still depends on the bandwidth [Duong, 2001].

Let X be a random viable with continuous distribution and density function $f(x) = \frac{d}{dx}F(x)$. A natural estimate of $F(x)$ is the empirical distribution function:

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{X_i \leq x\}}. \quad (\text{A.14})$$

Consider a discrete derivative:

$$\hat{f}(x) = \frac{\hat{F}(x+h) - \hat{F}(x-h)}{2h}, \quad \text{for some } h > 0. \quad (\text{A.15})$$

By definition of (A.14), the right side of the equation can also be written as:

$$\frac{1}{2hn} \sum_{i=1}^n \mathbb{1}_{\{x-h \leq X_i \leq x+h\}} = \frac{1}{2hn} \sum_{i=1}^n \mathbb{1}_{\{|\frac{X_i-x}{h}| \leq 1\}}, \quad (\text{A.16})$$

where the right side of the equation above can be written as :

$$\frac{1}{2hn} \sum_{i=1}^n \mathbb{1}_{\{|\frac{X_i-x}{h}| \leq 1\}} = \frac{1}{hn} \sum_{i=1}^n k\left(\frac{X_i-x}{h}\right),$$

where

$$k(u) = \begin{cases} \frac{1}{2} & |u| \leq 1 \\ 0 & |u| > 1, \end{cases} \quad (\text{A.17})$$

which is the uniform density function on $[-1, 1]$.

Therefore, the general form of the kernel density estimator is defined as:

$$\hat{f}(x) = \frac{1}{hn} \sum_{i=1}^n k\left(\frac{X_i-x}{h}\right), \quad (\text{A.18})$$

where $k(\cdot)$ is a non-negative function that integrates to 1. It is called a kernel smoothing function which sums the component smoothing functions for each data value to produce a smooth, continuous density. The parameter h is the bandwidth, which controls the degree of smoothing [Hansen, 2009, pp.2-3]. Possible choices of kernel functions are the normal, triangular, biweight, Gaussian ect.. Here we are using the Gaussian kernel $k_\phi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$.

A.5.1 Gaussian Kernel Method

Assume \hat{Z} to be the empirical distribution and define:

$$\hat{Z}_h = \hat{Z} + h\varepsilon, \quad (\text{A.19})$$

where $h \geq 0$, $\varepsilon \sim N(0, 1)$ and s is the sample standard deviation. Hence:

$$\Pr(\hat{Z}_h \leq z | \hat{Z} = z_i) = \Phi\left(\frac{z - z_i}{hs}\right). \quad (\text{A.20})$$

Thus,

$$\Pr(\hat{Z}_h \leq z) = \frac{1}{n} \sum_{i=1}^n \Phi\left(\frac{z - z_i}{hs}\right), \quad (\text{A.21})$$

where $\Phi(z) = \int_0^z f(y; 0, 1) dy$ is the standard normal integral. By differentiating, the density function $f(\hat{Z}_h)$ is given by:

$$f(\hat{z}_h) = \frac{1}{n} \sum_{i=1}^n \frac{1}{hs} \varphi\left(\frac{z - Z_i}{hs}\right), \quad \text{where } \varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}. \quad (\text{A.22})$$

A.6 Derivation of the Tail Distribution Function of Z

Recall that $Z_b = Z - b$ given $Z > b$. Then the tail distribution function $\bar{F}_b = \Pr(Z_b > z | Z > b)$ is given by:

$$\begin{aligned} \bar{F}_b(z) &= \frac{\Pr(Z_b > z, Z > b)}{\Pr(Z > b)} \\ &= \frac{\Pr(Z - b > z, Z > b)}{\Pr(Z > b)} \\ &= \frac{\Pr(Z > z + b)}{\Pr(Z > b)} \\ &= \frac{1 - \Pr(Z \leq z + b)}{1 - \Pr(Z \leq b)} \\ &= \frac{1 - F(z + b)}{1 - F(b)}, \quad \text{where } F(z) = \Pr(Z \leq z). \end{aligned} \quad (\text{A.23})$$

Recall that $Y_b = Z_b/\beta_b$, the tail distribution function $\Pr(Y_b > y)$ is given by:

$$\begin{aligned} \Pr(Y_b > y) &= \Pr\left(\frac{Z_b}{\beta_b} > y | Z > b\right) \\ &= \Pr\left(\frac{Z - b}{\beta_b} > y | Z > b\right) \\ &= \frac{\Pr\left(\frac{Z - b}{\beta_b} > y, Z > b\right)}{\Pr(Z > b)} \\ &= \frac{\Pr(Z > y\beta_b + b)}{\Pr(Z > b)} \\ &= \frac{1 - F(y\beta_b + b)}{1 - F(b)}. \end{aligned}$$

By (A.23), the tail distribution function $\Pr(Y_b > y) = \bar{F}_b(\beta_b y)$.

Appendix B

R-scripts

Codes for the catastrophe frequency modelling:

```
library(MASS)
library("laeken")
library(fitdistrplus)
library(actuar)
library(evir)
data=read.table("/Users/yuexu/Documents/Master Thesis/natur_dnb.txt")

day=data[[1]]
month=data[[2]]
year=data[[3]]
kommunenr=data[[4]]
skadetype=data[[5]]
event=data[[6]]
fasts=data[[7]]

aar <- unique(year)
kommuner <- sort(unique(kommunenr))
skadetyper <- sort(unique(skadetype))
antskader <-array(dim=c(length(kommuner),length(aar),length(skadetyper)))

for(i in 1:length(kommuner))
{
  indi <- which(kommunenr == kommuner[i])
  for(j in 1:length(aar))
  {
    indij <- which(year[indi] == aar[j])
    for(k in 1:length(skadetyper))
    {
      indijk <- which(skadetype[indi[indij]] == skadetyper[k])
      antskader[i,j,k] <- length(indijk)
    }
  }
}

#####
###Catastrophe frequency modelling based on all catastrophes###
#####
antskadertot <- apply(antskader,c(1,2),sum)
n <- length(antskadertot)
#
#1. Poisson distribution
#
lambda_tot <- sum(antskadertot)/n # 5.792002
p <- (1:n-0.5)/n
```

```

qpar <- qpois(p, lambda_tot)

plot(qpar, sort(antskadertot)) # QQ-plot

# Comparison of the probability density
phat <- c(sum(antskadertot == 0), tabulate(antskadertot))/n
plot(0:50, phat[1:51], pch='_', col="red")
points(0:50, dpois(0:50, lambda_tot), pch='_', col="blue")
legend(x="topright", legend=c("Total_claim_frequency", "Poisson_distr."), col=c("red", "blue"), lty=c(1,1))

#
#2. Negative binomial distribution
#
NBtotal <- fitdist(sort(antskadertot), "nbinom") # Maximum likelihood estimation
# Parameters:
#   estimate Std. Error
# size 1.069890 0.01495271
# mu   5.791611 0.05034247

xi_tot <- 5.791611
alpha_tot <- 1.069890
prob_tot <- alpha_tot/(alpha_tot+xi_tot) # 0.1559265
nbpar <- qnbinom(p, alpha_tot, prob_tot)

plot(nbpar, sort(antskadertot)) # QQ-plot

# Comparison of the probability density
plot(0:50, phat[1:51], pch='_', col="red")
points(0:50, dnbinom(0:50, alpha_tot, prob_tot), pch='_', col="blue")
legend(x="topright", legend=c("Total_claim_frequency", "Negative_binomial_distr."), col=c("red", "blue"), lty=c(1,1))

#####
###Catastrophe frequency based on catastrophe types###
#####
#####
###Type 1:###
#####
type1 <- antskader[1:431, 1:34, 1]
n1 <- length(type1)
#
#1. Poisson distribution
#
lambda1 <- sum(type1)/n1 # 4.056026
p1 <- (1:n1-0.5)/n1
qpar1 <- qpois(p1, lambda1)

plot(qpar1, sort(type1))

phat1=c(sum(type1 == 0), tabulate(type1))/n1
plot(0:50, phat1[1:51], pch='_', col="red")
points(0:50, dpois(0:50, lambda1), pch='_', col="blue")
legend(x="topright", legend=c("Claim_frequency_type1", "Poisson_distr."), col=c("red", "blue"), lty=c(1,1))

#
#2. Negative binomial distribution
#
NBtype1=fitdist(sort(type1), "nbinom") # MLE
# Parameters :
#   estimate Std. Error

```

```

# size 1.063813 0.01640653
# mu 4.055916 0.03649612

xi1 <- 4.055916
alpha1 <- 1.063813
prob1 <- alpha1/(alpha1+xi1) # 0.207787
nbpar1=qnbinom(p1,alpha1,prob1)

plot(nbpar1,sort(type1))

plot(0:50,phat1[1:51],pch='_',col="red")
points(0:50,dnbinom(0:50,alpha1,prob1),pch='_',col="blue")
legend(x="topright",legend=c("Claims_frequency_type1","Negative_binomial_distr."),col=c("red","blue"),lty=c
(1,1))

#####
###Type 2 and type 3 merged###
#####
type2 <- antskader[1:431,1:34,2]
type3 <- antskader[1:431,1:34,3]
type23 <- type2+type3
n23 <- length(type23)
#
#1. Poisson distribution
#
lambda23 <- sum(type23)/n23 # 1.323598
p23 <- (1:n23-0.5)/n23
qpar23 <- qpois(p23,lambda23)

plot(qpar23,sort(type23))

phat23=c(sum(type23 == 0),tabulate(type23))/n23
plot(0:45,phat23[1:46],pch='_',col="red")
points(0:45,dpois(0:45,lambda23),pch='_',col="blue")
legend(x="topright",legend=c("Claim_frequency_type2_and_type3","Poisson_distr."),col=c("red","blue"),lty=c
(1,1))

#
#2. Negative binomial distribution
#
NBtype23 <- fitdist(sort(type23),"nbinom")
# Parameters:
# estimate Std. Error
# size 0.4686309 0.009411731
# mu 1.3236635 0.018587379

xi23 <- 1.3236635
alpha23 <- 0.4686309
prob23 <- alpha23/(alpha23+xi23) # 0.2614698
nbpar23 <- qnbinom(p23,alpha23,prob23)

plot(nbpar23,sort(type23))

plot(0:40,phat23[1:41],pch='_',col="red")
points(0:40,dnbinom(0:40,alpha23,prob23),pch='_',col="blue")
legend(x="topright",legend=c("Claims_frequency_type2_and_type3","Negative_binomial_distr."),col=c("red","
blue"),lty=c(1,1))

```

```
#####
###Type 4, type 5 and type 9 merged###
#####
type4 <- antskader[1:431,1:34,4]
type5 <- antskader[1:431,1:34,5]
type9 <- antskader[1:431,1:34,6]
type459 <- type4+type5+type9
n459 <- length(type459)
#
#1. Poisson distribution
#
lambda459 <- sum(type459)/n459 # 0.4123789
p459 <- (1:n459-0.5)/n459
qpar459 <- qpois(p459,lambda459)

plot(qpar459,sort(type459))

phat459=c(sum(type459 == 0),tabulate(type459)/n459)
plot(0:35,phat459[1:36],pch='_',col="red")
points(0:35,dpois(0:35,lambda459),pch='_',col="blue")
legend(x="topright",legend=c("Claim_frequency_type4_type5_and_type9","Poisson_distr."),col=c("red","blue"),
      lty=c(1,1))

#
#2. Negative binomial distribution
#
NBtype459 <- fitdist(sort(type459),"nbinom")
# Parameters:
#   estimate Std. Error
# size 0.2812828 0.009034710
# mu 0.4125248 0.008335132

xi459=0.4125248
alpha459=0.2812828
prob459=alpha459/(alpha459+xi459) # 0.405419
nbpar459=qnbinom(p459,alpha459,prob459)

plot(nbpar459,sort(type459))

plot(0:35,phat459[1:36],pch='_',col="red")
points(0:35,dnbinom(0:35,alpha459,prob459),pch='_',col="blue")
legend(x="topright",legend=c("Claims_frequency_type4_type5_and_type9","Negative_binomial_distr."),col=c("
red","blue"),lty=c(1,1))
```

Codes for the amounts of compensation modelling

```
#####
###Compensation amounts modelling###
#####
#Remove claim sizes that are smaller than 3:
fasts_new <- fasts[fasts > 3]
probs <- c(seq(0,0.9,.05),seq(0.9,0.99,.001),seq(0.99,0.999,.0001))
quantile(fasts_new,probs=probs)

meanExcessPlot(fasts_new,probs=probs,interactive=TRUE)
abline(v=13898075,col="red")

#From this point the observations form almost a straight line:
#$x0
#[1] 13898075

#$k
```

```

#[1] 113 # Number of observations over the threshold

#attr(,"class")
#[1] "paretoScale"

#Find the corresponding quantile
fhat <- ecdf(fasts_new)
fhat(13898075)
#[1] 0.9984016 # 0.2% over-threshold part

fasts_new_c <- fasts_new[fasts_new <= 13898075] # Central part data

#####
###Central part###
#####
#
#1. Log-normal distribution
#
n_tot <- length(fasts_new_c) # 70582
mu_tot <- mean(log(fasts_new_c)) # 9.955264
sigma_tot <- sd(log(fasts_new_c)) # 1.770335
xi_lognorm_tot <- exp(sigma_tot^2/2+mu_tot) # 100942.6

p <- (1:n_tot-0.5)/n_tot
qpar <- qlnorm(p,mu_tot,sigma_tot)

plot(qpar,sort(fasts_new_c)) # QQ-plot

fhat_tot <- density(fasts_new_c,from=0,to=3e5) # Density of the data
# Comparison of the density function:
plot(fhat_tot,type="l",xlab="",ylab="",main=NA,xlim=c(0,3e05),ylim=c(0,5.5e-5),col="red")
lines(1:3e5,dlnorm(1:3e5,mu_tot,sigma_tot),main=NA,col="blue")
legend(x="topright",legend=c("Claim_size_all_central","Log_normal_distr."),col=c("red","blue"),lty=c(1,1))

#
#2.Gamma distribution
#
llminus <- function(alpha,z)-alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z))) # Log-
likelihood function
xihat <- mean(fasts_new_c) # 131928.6
optimal <- optimize(llminus,c(1,1e20)*xihat/sd(fasts_new_c),z=fasts_new_c)
alphahat <- optimal[[1]] # 0.3641549
p <- (1:n_tot-0.5)/n_tot
qgam <- xihat*qgamma(p,alphahat)/alphahat
Zgam <- xihat*rgamma(n_tot,alphahat)/alphahat

plot(qgam,sort(fasts_new_c)) # QQ-plot

#Comparison of the density
plot(fhat_tot,type="l",xlab="",ylab="",main=NA,ylim=c(0,4e-5),col="red")
lines(density(Zgam,from=0),main=NA,col="blue")
legend(x="topright",legend=c("Claim_size_all_central","Gamma_distr."),col=c("red","blue"),lwd=1,lty=c(1,1))

#Try transformation by taking logarithm to the data
#
#2.1 Log-Gamma distribution
#
llminus <- function(alpha,z)-alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z)))
xihat_log <- mean(log(fasts_new_c)) # 9.955264
optimal <- optimize(llminus,c(1,1e20)*xihat_log/sd(log(fasts_new_c)),z=log(fasts_new_c))
alphahat_log=optimal[[1]] # 31.25975
p <- (1:n_tot-0.5)/n_tot
qgam_log <- xihat_log*qgamma(p,alphahat_log)/alphahat_log
Zgam_log <- xihat_log*rgamma(n_tot,alphahat_log)/alphahat_log

```



```

plot(qgam_log, sort(log(fasts_new_c)))

fhat_log <- density(log(fasts_new_c), from=0, to=20) # Density of the log data
plot(fhat_log, type="l", xlab="", ylab="", main=NA, col="red")
lines(density(Zgam_log, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_all_central_log", "Gamma_distr."), col=c("red", "blue"), lty=c(1,1))

#
#3. Pareto distribution
#
llminus <- function(beta, z)
{
  alphahatbeta=1/mean(log(1+z/beta));
  -log(alphahatbeta/beta) + (1+1/alphahatbeta)
}
o <- optimize(llminus, c(1, 1e20), z=fasts_new_c);
betahat <- o[[1]]; # 9.108538e+19
alphahat <- 1/mean(log(1+fasts_new_c/betahat)) # 6.956652e+14
p <- (1:n_tot-0.5)/n_tot;
qpar <- betahat*((1-p)**(-1/alphahat))-1
Zpar <- betahat*(runif(n_tot)**(-1/alphahat))-1

plot(qpar, sort(fasts_new_c))

plot(fhat_tot, type="l", xlab="", ylab="", main=NA, ylim=c(0, 3.5e-5), col="red")
lines(density(Zpar, from=0), col="blue")
#lines(1:5e5, dpareto(1:5e5, alphahat, betahat), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_all_central", "Pareto_distr."), col=c("red", "blue"), lwd=1, lty=c(1,1))
)

#
#4. Extended Pareto distribution
#
llminus <- function(s, z){t=exp(s);
l_1=-lgamma(t[1]+t[2])+lgamma(t[1])+lgamma(t[2])+t[2]*log(t[3]);
l_2=-(t[2]-1)*mean(log(z))+(t[1]+t[2])*mean(log(1+z/t[3]));
l_1+l_2;
}
o <- optim(c(1, 0.7, log(mean(fasts_new_c))), llminus, z=fasts_new_c)
alpha_expar <- exp(o$par[1]) # 0.8557846
theta_expar <- exp(o$par[2]) # 1.257528
beta_expar <- exp(o$par[3]) # 11271.97

W <- beta_expar*rgamma(1e5, theta_expar)/rgamma(1e5, alpha_expar)
b <- 13898075
ind <- which(W > b) # Count the indices over the threshold
W <- W[-ind] # Data under the threshold
Z <- W[1:n_tot]

plot(sort(Z), sort(fasts_new_c))

dExPareto <- function(z, alpha, theta, beta)
{
  gamma(alpha+theta)/(gamma(alpha)*gamma(theta))*z^(theta-1)/(beta^theta*(1+z/beta)^(alpha+theta))
} # Density function of the extended Pareto

z <- 1:(3e5)
plot(fhat_tot, type="l", xlab="", ylab="", ylim=c(0, 4.5e-05), main=NA, col="red")
lines(z, dExPareto(z, alpha_expar, theta_expar, beta_expar), col="blue")
legend(x="topright", legend=c("Claim_size_all_central", "Extended_Pareto_distr."), col=c("red", "blue"), lwd=1,
lty=c(1,1))

```

```

#####
###Over-threshold part###
#####
fasts_new_e <- fasts_new[fasts_new > 13898075]-13898075 # Over-threshold data
#
#Pareto distribution
#
llminus <- function(beta,z)
{
  alphahatbeta=1/mean(log(1+z/beta));
  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
}
o <- optimize(llminus,c(1,1e12),z=fasts_new_e);
betahat_e <- o[[1]]; # 19350799
alphahat_e <- 1/mean(log(1+fasts_new_e/betahat_e)) # 2.119329
n<- length(fasts_new_e); # 113
p <- (1:n-0.5)/n;
qpar_e <- betahat_e*((1-p)**(-(1/alphahat_e))-1)
Zpar_e <- betahat_e*(runif(n)**(-(1/alphahat_e))-1)

plot(qpar_e,sort(fasts_new_e))

plot(density(fasts_new_e,from=0),main=NA,col="red",ylim=c(0,6.5e-08))
lines(density(Zpar_e,from=0),main=NA,col="blue")
legend(x="topright",legend=c("Claim_size_all_over_threshold","Pareto_distr."),col=c("red","blue"),lwd=1,ltty
      =c(1,1))

#####
###Claim size of type 1###
#####
new_claim1 <- subset(data,fasts>3)
fasts1 <- new_claim1[new_claim1$V5==1,]
new_fasts_tp1 <- fasts1[[7]]

probs <- c(seq(0,0.9,.05),seq(0.9,0.99,.001),seq(0.99,0.999,.0001))
quantile(new_fasts_tp1,probs=probs)

meanExcessPlot(new_fasts_tp1,probs=probs,interactive=TRUE)
abline(v=6903315,col="red")

#From this point the observations form almost a straight line:
#$x0
#[1] 6903315

#$k
#[1] 137 # Number of observations over the threshold

#attr(,"class")
#[1] "paretoScale"

#Find the corresponding quantile
Fhat_tp1 <- ecdf(new_fasts_tp1)
Fhat_tp1(6903315)
#[1] 0.9973011 # 0.3% over-threshold

fasts_tp1_c <- new_fasts_tp1[new_fasts_tp1<=6903315] # Central part data

```

```

#####
##Central part##
#####
#
#1. Log-normal distribution
#
n1 <- length(fasts_tp1_c) # 50625
mul <- mean(log(fasts_tp1_c)) # 9.852738
sigma1 <- sd(log(fasts_tp1_c)) # 1.686393
xi_lognorm1 <- exp(sigma1^2/2+mul) # 78802.36

p1 <- (1:n1-0.5)/n1
fhat1 <- density(fasts_tp1_c, from=0, to=3e5)
qpar1 <- qlnorm(p1, mul, sigma1)

plot(qpar1, sort(fasts_tp1_c))

plot(fhat1, type="l", xlab="", ylab="", main=NA, xlim=c(0, 250000), ylim=c(0, 5.5e-5), col="red")
lines(1:3e5, dlnorm(1:3e5, mul, sigma1), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_typed1_central", "Log_normal_distr."), col=c("red", "blue"), lty=c(1, 1)
)

#
#2. Gamma distribution
#
llminus <- function(alpha, z) -alpha*(log(alpha)-1) + lgamma(alpha) + alpha*(log(mean(z)) - mean(log(z)))
xihat1 <- mean(fasts_tp1_c) # 93638.03
optimal <- optimize(llminus, c(1, 1e12) * xihat1 / sd(fasts_tp1_c), z = fasts_tp1_c)
alphahat1 <- optimal[[1]] # 0.4109333
p1 <- (1:n1-0.5)/n1
qgam1 <- xihat1 * qgamma(p1, alphahat1) / alphahat1
Zgam1 <- xihat1 * rgamma(n1, alphahat1) / alphahat1

plot(qgam1, sort(fasts_tp1_c))

plot(fhat1, type="l", xlab="", ylab="", main=NA, ylim=c(0, 4e-5), col="red")
lines(density(Zgam1, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_typed1_central", "Gamma_distr."), col=c("red", "blue"), lty=c(1, 1))

#
#2.1 Log-Gamma distribution
#
llminus <- function(alpha, z) -alpha*(log(alpha)-1) + lgamma(alpha) + alpha*(log(mean(z)) - mean(log(z)))
xihat1_log <- mean(log(fasts_tp1_c)) # 9.852738
optimal <- optimize(llminus, c(1, 1e12) * xihat1_log / sd(log(fasts_tp1_c)), z = log(fasts_tp1_c))
alphahat1_log <- optimal[[1]] # 33.54351
p1 <- (1:n1-0.5)/n1
qgam1_log <- xihat1_log * qgamma(p1, alphahat1_log) / alphahat1_log
Zgam1_log <- xihat1_log * rgamma(n1, alphahat1_log) / alphahat1_log

plot(qgam1_log, sort(log(fasts_tp1_c)))

fhat1_log <- density(log(fasts_tp1_c), from=0, to=20)
plot(fhat1_log, type="l", xlab="", ylab="", ylim=c(0, 0.28), main=NA, col="red")
lines(density(Zgam1_log, from=0), col="blue")
legend(x="topright", legend=c("Claim_size_typed1_central_log", "Gamma_distr."), col=c("red", "blue"), lty=c(1, 1))

#
#3. Pareto distribution
#
llminus <- function(beta, z)
{
  alphahatbeta = 1 / mean(log(1+z/beta));
  -log(alphahatbeta/beta) + (1+1/alphahatbeta)
}

```

```

}
o <- optimize(llminus,c(1,1e20),z=fasts_tpl_c);
betahat1 <- o[[1]]; # 9.923929e+19
alphahat1 <- 1/mean(log(1+fasts_tpl_c/betahat1)) # 1.072477e+15
p1 <- (1:n1-0.5)/n1;
qpar1 <- betahat1*((1-p1)**(-(1/alphahat1))-1)
Zpar1 <- betahat1*(runif(n1)**(-(1/alphahat1))-1)

plot(qpar1,sort(fasts_tpl_c))

plot(fhat1,type="l",xlab="",ylab="",main=NA,ylim=c(0,4e-5),col="red")
lines(density(Zpar1,from=0,bw=20000),main=NA,col="blue")
legend(x="topright",legend=c("Claim_size_type1_central","Pareto_distr."),col=c("red","blue"),lwd=1,lty=c
(1,1))

#
#4. Extended Pareto distribution
#
llminus <- function(s,z){t=exp(s);
l_1=-lgamma(t[1]+t[2])+lgamma(t[1])+lgamma(t[2])+t[2]*log(t[3]);
l_2=-(t[2]-1)*mean(log(z))+(t[1]+t[2])*mean(log(1+z/t[3]));
l_1+l_2;
}
o <- optim(c(1,0.7,log(mean(fasts_tpl_c))),llminus,z=fasts_tpl_c)
alpha1_expar <- exp(o$par[1]) # 0.9464847
theta1_expar <- exp(o$par[2]) # 1.27438
beta1_expar <- exp(o$par[3]) # 11881.65

W1=beta1_expar*rgamma(1e5,theta1_expar)/rgamma(1e5,alpha1_expar)
b1 <- 6903315
ind1 <- which(W1 > b1)
W1 <- W1[-ind1]
Z1 <- W1[1:n1]

plot(sort(Z1),sort(fasts_tpl_c))

z1 <- 1:(3e5)
plot(fhat1,type="l",xlab="",ylab="",main=NA,ylim=c(0,5e-5),col="red")
lines(z1,dExPareto(z1,alpha1_expar,theta1_expar,beta1_expar),col="blue",main=NA)
legend(x="topright",legend=c("Claim_size_type1_central","Extended_Pareto_distr."),col=c("red","blue"),lwd
=1,lty=c(1,1))

#####
###Over-threshold part###
#####
fasts_tpl_e <- new_fasts_tpl[new_fasts_tpl>6903315]-6903315
#
#Pareto
#
llminus <- function(beta,z)
{
  alphahatbeta=1/mean(log(1+z/beta));
  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
}
o <- optimize(llminus,c(1,1e12),z=fasts_tpl_e);
betahat1_e <- o[[1]]; # 38509614
alphahat1_e <- 1/mean(log(1+fasts_tpl_e/betahat1_e)) # 4.636787
n <- length(fasts_tpl_e);
p <- (1:n-0.5)/n;
qpar1_e <- betahat1_e*((1-p)**(-(1/alphahat1_e))-1)
Zpar1_e <- betahat1_e*(runif(n)**(-(1/alphahat1_e))-1)

```

```

plot(qpar1_e, sort(fasts_tp1_e))

plot(density(fasts_tp1_e, from=0), col="red", ylim=c(0, 9e-08), main=NA)
lines(density(2par1_e, from=0), col="blue", main=NA)
legend(x="topright", legend=c("Claim_size_type1_over_threshold", "Pareto_distr."), col=c("red", "blue"), lwd=1,
      lty=c(1,1))

#####
###Claim size of type2 and typ3###
#####
new_claim <- subset(data, fasts>3)
fasts2 <- new_claim[new_claim$V5==2,]
new_fasts_tp2 <- fasts2[[7]]
fasts3 <- new_claim[new_claim$V5==3,]
new_fasts_tp3 <- fasts3[[7]]
new_fasts_tp23 <- c(new_fasts_tp2, new_fasts_tp3)

probs <- c(seq(0, 0.9, .05), seq(0.9, 0.99, .001), seq(0.99, 0.999, .0001))
quantile(new_fasts_tp23, probs=probs)

meanExcessPlot(new_fasts_tp23, probs=probs, interactive=TRUE)
abline(v=15108527, col="red")

#From this point the observations form almost a straight line:
#$x0
#[1] 15108527

#$k
#[1] 36 # Number of observations over the threshold

#attr("class")
#[1] "paretoScale"

#Find the corresponding quantile
Fhat23 <- ecdf(new_fasts_tp23)
Fhat23(15108527)
#0.9977321 # 0.2% over-threshold

fasts_tp23_c <- new_fasts_tp23[new_fasts_tp23<=15108527]
#####
###Central part###
#####
#
#1. Log-normal distribution
#
n23 <- length(fasts_tp23_c) # 15838
mu23 <- mean(log(fasts_tp23_c)) # 10.26194
sigma23 <- sd(log(fasts_tp23_c)) # 1.893885
xi_lognorm23 <- exp(sigma23^2/2+mu23) # 172016

p23 <- (1:n23-0.5)/n23
qpar23 <- qlnorm(p23, mu23, sigma23)
fhat23 <- density(fasts_tp23_c, from=0, to=3e5)

plot(qpar23, sort(fasts_tp23_c))

plot(fhat23, type="l", xlab="", ylab="", main=NA, xlim=c(0, 250000), ylim=c(0, 5e-5), col="red")
lines(1:3e5, dlnorm(1:3e5, mu23, sigma23), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_type2_and_type3_central", "Log_normal_distr."), col=c("red", "blue"),
      lty=c(1,1))

```

```

#
#2. Gamma distribution
#
llminus <- function(alpha, z)-alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z)))
xihat23 <- mean(fasts_tp23_c) # 205825.5
optimal <- optimize(llminus, c(1, 1e12)*xihat23/sd(fasts_tp23_c), z=fasts_tp23_c)
alphahat23 <- optimal[[1]] # 0.3421276
p23 <- (1:n23-0.5)/n23
qgam23 <- xihat23*qgamma(p23, alphahat23)/alphahat23
Zgam23 <- xihat23*rgamma(n23, alphahat23)/alphahat23

plot(qgam23, sort(fasts_tp23_c))

plot(fhat23, type="l", xlab="", ylab="", ylim=c(0, 2.5e-05), main=NA, col="red")
lines(density(Zgam23, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_type2_and_type3_central", "Gamma_distr."), col=c("red", "blue"), lty=c
(1, 1))

#
#2.1 Log-Gamma distribution
#
llminus <- function(alpha, z)-alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z)))
xihat23_log <- mean(log(fasts_tp23_c)) # 10.26194
optimal <- optimize(llminus, c(1, 1e12)*xihat23_log/sd(log(fasts_tp23_c)), z=log(fasts_tp23_c))
alphahat23_log <- optimal[[1]] # 28.81373
p23 <- (1:n23-0.5)/n23
qgam23_log <- xihat23_log*qgamma(p23, alphahat23_log)/alphahat23_log
Zgam23_log <- xihat23_log*rgamma(n23, alphahat23_log)/alphahat23_log

plot(qgam23_log, sort(log(fasts_tp23_c)))

fhat23_log <- density(log(fasts_tp23_c), from=0, to=20)
plot(fhat23_log, type="l", xlab="", ylab="", ylim=c(0, 0.25), main=NA, col="red")
lines(density(Zgam23_log, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_type2_and_type3_central_log", "Gamma_distr."), col=c("red", "blue"),
lty=c(1, 1))

#
#3. Pareto distribution
#
llminus <- function(beta, z)
{
  alphahatbeta <- 1/mean(log(1+z/beta));
  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
}
o <- optimize(llminus, c(1, 1e20), z=fasts_tp23_c);
betahat23 <- o[[1]]; # 9.585355e+19
alphahat23 <- 1/mean(log(1+fasts_tp23_c/betahat23)) # 4.678934e+14
qpar23 <- betahat23*((1-p23)**(-(1/alphahat23)))-1
Zpar23 <- betahat23*(runif(n23)**(-(1/alphahat23)))-1

plot(qpar23, sort(fasts_tp23_c))

plot(fhat23, type="l", xlab="", ylab="", main=NA, ylim=c(0, 2e-5), col="red")
lines(density(Zpar23, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_type2_and_type3_central", "Pareto_distr."), col=c("red", "blue"), lwd
=1, lty=c(1, 1))

#
#4. Extended Pareto distribution
#
llminus <- function(s, z){t=exp(s);
l_1=-lgamma(t[1]+t[2])+lgamma(t[1])+lgamma(t[2])+t[2]*log(t[3]);

```

```

l_2=- (t[2]-1)*mean(log(z))+(t[1]+t[2])*mean(log(1+z/t[3]));
l_1+l_2;
}
o <- optim(c(1,0.7,log(mean((fasts_tp23_c)))) ,llminus,z=fasts_tp23_c)
alpha23_expar <- exp(o$par[1]) # 0.798279
theta23_expar <- exp(o$par[2]) # 1.083934
beta23_expar <- exp(o$par[3]) # 16987.06

W23 <- beta23_expar*rgamma(1e5,theta23_expar)/rgamma(1e5,alpha23_expar)
b23 <- 15108527
ind23 <- which(W23 > b23)
W23<- W23[-ind23]
Z23<- W23[1:n23]

plot(sort(Z23),sort(fasts_tp23_c))

z23 <- 1:(3e5)
plot(fhat23,type="l",xlab="",ylab="",main=NA,ylim=c(0,4e-5),col="red")
lines(z23,dExPareto(z23,alpha23_expar,theta23_expar,beta23_expar),col="blue",main=NA)
legend(x="topright",legend=c("Claim_size_type2_and_type3_central","Extended_Pareto_distr."),col=c("red","blue"),lwd=1,lty=c(1,1))

#####
###Over-threshold part###
#####
fasts_tp23_e <- new_fasts_tp23[new_fasts_tp23>15108527]-15108527
#
#Pareto
#
llminus <- function(beta,z)
{
  alphahatbeta=1/mean(log(1+z/beta));
  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
}
o <- optimize(llminus,c(1,1e12),z=fasts_tp23_e);
betahat23_e <- o[[1]]; # 34391349
alphahat23_e <- 1/mean(log(1+fasts_tp23_e/betahat23_e)) # 2.446868
n <- length(fasts_tp23_e); # 36
p <- (1:n-0.5)/n;
qpar23_e <- betahat23_e*((1-p)**(-(1/alphahat23_e))-1)
Zpar23_e <- betahat23_e*(runif(n)**(-(1/alphahat23_e))-1)

plot(qpar23_e,sort(fasts_tp23_e))

plot(density(fasts_tp23_e,from=0),ylim=c(0,4e-08),col="red",main=NA)
lines(density(Zpar23_e,from=0),col="blue",main=NA)
legend(x="topright",legend=c("Claim_size_type2_and_type3_over_threshold","Pareto_distr."),col=c("red","blue"),lwd=1,lty=c(1,1))

#####
###Claim size of type 4, type 5 and type 9###
#####
new_claim <- subset(data,fasts>3)
fasts4 <- new_claim[new_claim$V5==4,]
new_fasts_tp4 <- fasts4[[7]]
fasts5 <- new_claim[new_claim$V5==5,]
new_fasts_tp5 <- fasts5[[7]]
fasts9 <- new_claim[new_claim$V5==9,]
new_fasts_tp9 <- fasts9[[7]]

```

```

new_fasts_tp459 <- c(new_fasts_tp4,new_fasts_tp5,new_fasts_tp9) # 4059

probs <- c(seq(0,0.9,.05),seq(0.9,0.99,.001),seq(0.99,0.999,.0001))
quantile(new_fasts_tp459,probs=probs)

meanExcessPlot(new_fasts_tp459,probs=probs,interactive=TRUE)
abline(v=8876982,col="red")

#From this point the observations form almost a straight line:
#x0
#[1] 8876982

#k
#[1] 9 # Number of observations over the threshold

#attr("class")
#[1] "paretoScale"

#Find the corresponding quantile
Fhat459 <- ecdf(new_fasts_tp459)
Fhat459(8876982)
#0.9977827 # 0.2% over-threshold

fasts_tp459_c <- new_fasts_tp459[new_fasts_tp459<=8876982]

#####
###Central part:###
#####
#
#1. Log-normal distribution
#
n459 <- length(fasts_tp459_c) # 4050
mu459 <- mean(log(fasts_tp459_c)) # 9.933211
sigma459 <- sd(log(fasts_tp459_c)) # 1.976607
xi_lognorm459 <- exp(sigma459^2/2+mu459) # 145320.7

p459 <- (1:n459-0.5)/n459
qpar459 <- qlnorm(p459,mu459,sigma459)
fhat459 <- density(fasts_tp459_c,from=0,to=3e5)

plot(qpar459,sort(fasts_tp459_c))

plot(fhat459,type="l",xlab="",ylab="",main=NA,xlim=c(0,200000),ylim=c(0,7e-5),col="red")
lines(1:3e5,dlnorm(1:3e5,mu459,sigma459),main=NA,col="blue")
legend(x="topright",legend=c("Claim_size_type4_type5_and_type9_central","Log_normal_distr."),col=c("red","blue"),lty=c(1,1))

#
#2. Gamma distribution
#
llminus <- function(alpha,z)-alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z)))
xihat459 <- mean(fasts_tp459_c) # 157480.4
optimal <- optimize(llminus,c(1,1e12))*xihat459/sd(fasts_tp459_c),z=fasts_tp459_c)
alphahat459 <- optimal[[1]] # 0.3332724
p459 <- (1:n459-0.5)/n459
qgam459 <- xihat459*ggamma(p459,alphahat459)/alphahat459
Zgam459 <- xihat459*rgamma(n459,alphahat459)/alphahat459

plot(qgam459,sort(fasts_tp459_c))

plot(fhat459,type="l",xlab="",ylab="",ylim=c(0,2.5e-05),main=NA,col="red")
lines(density(Zgam23,from=0),main=NA,col="blue")
legend(x="topright",legend=c("Claim_size_type4_type5_and_type9_central","Gamma_distr."),col=c("red","blue"))

```



```

, lty=c(1,1))

#
#2.1 Log-Gamma distribution
#
llminus <- function(alpha, z) -alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z)))
xihat459_log <- mean(log(fasts_tp459_c)) # 9.933211
optimal <- optimize(llminus, c(1, 1e12)*xihat459_log/sd(log(fasts_tp459_c)), z=log(fasts_tp459_c))
alphahat459_log <- optimal[[1]] # 25.24475
qgam459_log <- xihat459_log*qgamma(p459, alphahat459_log)/alphahat459_log
Zgam459_log <- xihat459_log*rgamma(n459, alphahat459_log)/alphahat459_log

plot(qgam459_log, sort(log(fasts_tp459_c)))

fhat459_log <- density(log(fasts_tp459_c), from=0, to=20)
plot(fhat459_log, type="l", xlab="", ylab="", ylim=c(0, 0.25), main=NA, col="red")
lines(density(Zgam459_log, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_type4_type5_and_type9_central_log", "Gamma_distr."), col=c("red", "blue"), lty=c(1,1))

#
#3. Pareto distribution
#
llminus <- function(beta, z)
{
  alphahatbeta=1/mean(log(1+z/beta));
  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
}
o <- optimize(llminus, c(1, 1e20), z=fasts_tp459_c);
betahat459 <- o[[1]]; # 9.216166e+19
alphahat459 <- 1/mean(log(1+fasts_tp459_c/betahat459)) # 5.900676e+14
qpar459 <- betahat459*((1-p459)**(-(1/alphahat459))-1)
Zpar459 <- betahat459*(runif(10000)**(-(1/alphahat459))-1)

plot(qpar459, sort(fasts_tp459_c))

plot(fhat459, type="l", xlab="", ylab="", main=NA, ylim=c(0, 2.5e-5), col="red")
lines(density(Zpar459, from=0), main=NA, col="blue")
legend(x="topright", legend=c("Claim_size_type4_type5_and_type9_central", "Pareto_distr."), col=c("red", "blue"), lwd=1, lty=c(1,1))

#
#4. Extended Pareto distribution
#
llminus <- function(s, z) {t=exp(s);
l_1=-lgamma(t[1]+t[2])+lgamma(t[1])+lgamma(t[2])+t[2]*log(t[3]);
l_2=-(t[2]-1)*mean(log(z))+t[1]+t[2]*mean(log(1+z/t[3]));
l_1+l_2;
}
o <- optim(c(1, 0.7, log(mean(fasts_tp459_c))), llminus, z=fasts_tp459_c)
alpha459_expar <- exp(o$par[1]) # 0.6263603
theta459_expar <- exp(o$par[2]) # 1.4741
beta459_expar <- exp(o$par[3]) # 4778.121

W459 <- beta459_expar*rgamma(1e5, theta459_expar)/rgamma(1e5, alpha459_expar)
b459 <- 8876982
ind459 <- which(W459 > b459)
W459 <- W459[-ind459]
Z459 <- W459[1:n459]

plot(sort(Z459), sort(fasts_tp459_c))

z459 <- 1:(3e5)
plot(fhat459, type="l", xlab="", ylab="", main=NA, xlim=c(0, 2.5e5), ylim=c(0, 5.7e-5), col="red")

```

```

lines(z459,dExPareto(z459,alpha459_expar,theta459_expar,beta459_expar),col="blue",main=NA)
legend(x="topright",legend=c("Claim_size_type4_type5_and_type9_central","Extended_Pareto_distr."),col=c("
red","blue"),lwd=1,lty=c(1,1))

#####
###Over-threshold part:###
#####
fasts_tp459_e <- new_fasts_tp459[new_fasts_tp459>8876982]-8876982
#
#Pareto distribution
#
llminus <- function(beta,z)
{
  alphahatbeta=1/mean(log(1+z/beta));
  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
}
o <- optimize(llminus,c(1,1e12),z=fasts_tp459_e);
betahat459_e <- o[[1]]; # 999990662494
alphahat459_e <- 1/mean(log(1+fasts_tp459_e/betahat459_e)) # 42783.79
n <- length(fasts_tp459_e);
p <- (1:n-0.5)/n;
qpar459_e <- betahat459_e*((1-p)**(-(1/alphahat459_e))-1)
Zpar459_e <- betahat459_e*(runif(n)**(-(1/alphahat459_e))-1)

plot(qpar459_e,sort(fasts_tp459_e))

plot(density(fasts_tp459_e,from=0),ylim=c(0,4.5e-08),col="red",main=NA)
lines(density(Zpar459_e,from=0),col="blue",main=NA)
legend(x="topright",legend=c("Claim_size_type4_type5_and_type9_over_threshold","Pareto_distr."),col=c("red"
,"blue"),lwd=1,lty=c(1,1))

```

Codes for the bootstrap and Monte Carlo:

```

mb <- 1000 # Bootstrap replication
m <- 100000 # Number of MC simulation
epsilon <- 0.01

J <- 431
K <- 34

#####
###All data###
#####
#
#Bootstrap of the estimates from the negative binomial distribution
#
alpha_tot <- 1.069890 # ML estimate
xi_tot <- 5.791611 # ML estimate
n <- J*K
NB_tot <- matrix(rnegbin(n*mb,mu=xi_tot,theta=alpha_tot),n,mb)

llminusNBin <- function(alpha,n)
{
  xi=mean(n)
  -mean(lgamma(n+alpha))+lgamma(alpha)-alpha*log(alpha)-mean(n*log(xi)-(n+alpha)*log(alpha+xi))
}

xihatstar_NB <- 1:mb*0

```

```

alphahatstar_NB <- 1:mb*0

for (i in 1:mb)
{
  xihatstar_NB[i]=mean(NB_tot[,i])
  optimal_star=optimize(llminusNBin,c(0.0000001,100),n=NB_tot[,i])$min
  alphahatstar_NB[i]=optimal_star
}

#Confidence intervals for the estimates:
xihatstar_NB_CI <- c(quantile(xihatstar_NB,0.025),quantile(xihatstar_NB,0.975))
#(5.690591 5.897511)

alphahatstar_NB_CI <- c(quantile(alphahatstar_NB,0.025),quantile(alphahatstar_NB,0.975))
#(1.040933 1.101031)

#
#Bootstrap of the estimates from the Gamma distribution on log scale for
#the central part and from the Pareto for the extreme part
#
n_tot <- length(fasts_new)
b <- 13898075
rho <- 0.002 # 0.02% over-threshold part.
xihat_log <- 9.955264
alphahat_log <- 31.25975 # ML estimate
betahat_e <- 19350799
alphahat_e <- 2.119329

U <- matrix(runif(n_tot*mb),n_tot,mb)

llminusLGamma <- function(alpha,z)-alpha*(log(alpha)-1)+lgamma(alpha)+alpha*(log(mean(z))-mean(log(z))) #
  Log-likelihood function of the Gamma distribution

llminusPareto <- function(beta,z)
{
  alphahatbeta=1/mean(log(1+z/beta));

  -log(alphahatbeta/beta)+(1+1/alphahatbeta)
} # Log-likelihood function of the Pareto distribution

alphahatstar_gamma <- 1:mb*0
xihatstar_gamma <- 1:mb*0
betahatstar_e <- 1:mb*0
alphahatstar_e <- 1:mb*0

for (i in 1:mb)
{
  n_c=sum(U[,i] > rho)
  Zstar_c_log=xihat_log*rgamma(n_c,alphahat_log)/alphahat_log
  xihatstar_gamma[i]=mean(Zstar_c_log)
  optimal_star=optimize(llminusLGamma,c(1,1e20)*xihatstar_gamma[i]/sd(Zstar_c_log),z=Zstar_c_log)
  alphahatstar_gamma[i]=optimal_star[[1]]
  Zstar_e=betahat_e*(runif(n_tot-n_c)**(-1/alphahat_e)-1)
  optimalstar_e=optimize(llminusPareto, c(1,1e12),z=Zstar_e)
  betahatstar_e[i]=optimalstar_e[[1]]
  alphahatstar_e[i]=1/mean(log(1+Zstar_e/betahatstar_e[i]))
}

#Confidence intervals of the parameters
xihatstar_gamma_CI <- c(quantile(xihatstar_gamma,0.025),quantile(xihatstar_gamma,0.975))
#(9.942354 9.967597)

```

```

alphahatstar_gamma_CI <- c(quantile(alphahatstar_gamma,0.025),quantile(alphahatstar_gamma,0.975))
#(30.94986 31.60475)

alphahatstar_e_CI <- c(quantile(alphahatstar_e,0.025),quantile(alphahatstar_e,0.975))
#(1.449022 4.687899 )

betahatstar_e_CI <- c(quantile(betahatstar_e,0.025),quantile(betahatstar_e,0.975))
#(10746208 54251868)

#
#MC simulation of the reserve based on the common model: (w/o bootstrap)
#
Nhat <- rnegbin(m,mu=J*xi_tot,theta=J*alpha_tot)
Xhat <- 1:m*0

for (i in 1:m)
{
  U=runif(Nhat[i])
  n_c=sum(U>rho)
  n_e=sum(U<=rho)
  Z_c=exp(xihat_log*rgamma(n_c,alphahat_log)/alphahat_log)
  Z_e=b+betahat_e*(runif(n_e)**(-1/alphahat_e)-1)
  Xhat[i]=sum(Z_c)+sum(Z_e)
}
qhat <- sort(Xhat)[(1-epsilon)*m]
#1202608596

#
#Nested bootstrap Monte Carlo, negative binomial, log-Gamma and Pareto
#
m1<- m*mb
Xhatstar <- 1:m1*0
ind <- rep(1:mb,m) # replicate 1 to mb for m times
Nhatstar <- rnegbin(m1,mu=J*xihatstar_NB,theta=J*alphahatstar_NB)

for (j in 1:m1)
{
  U=runif(Nhatstar[j])
  n_c=sum(U>rho)
  n_e=sum(U<=rho)
  Z_c=exp(xihatstar_gamma[ind[j]]*rgamma(n_c,alphahatstar_gamma[ind[j]])/alphahatstar_gamma[ind[j]])
  Z_e=b+betahatstar_e[ind[j]]*(runif(n_e)**(-1/alphahatstar_e[ind[j]])-1)
  Xhatstar[j]=sum(Z_c)+sum(Z_e)
}
Xhatstar <- matrix(Xhatstar,byrow=T,m)
qhat_star <- apply(Xhatstar,2,sort)[m*(1-epsilon),]

#Percentile confidence interval for the reserve estimate
qhat_star_CI <- c(quantile(qhat_star,0.025),quantile(qhat_star,0.975))
#(1029884064 1677384667)

#Standard deviation of qhat_star
qhatstar_sd <- sd(qhat_star)
#195428602

#####
###Type 1###
#####
#
#Bootstrap of the estimates of NB

```

```

#
alpha1 <- 1.063813
x11 <- 4.055916

NB1<- matrix(rnegbin(n*mb,mu=x11,theta=alpha1),n,mb)

xihatstar1_NB <- 1:mb*0
alphahatstar1_NB <- 1:mb*0

for (i in 1:mb)
{
  xihatstar1_NB[i]=mean(NB1[,i])
  optimall_star=optimize(llminusNBin,c(0.0000001,100),n=NB1[,i])$min
  alphahatstar1_NB[i]=optimall_star
}

#
#Bootstrap of Gamma on log scale for the central part and Pareto for the extreme part:
#
n1 <- length(new_fasts_tpl)
b1 <- 6903315
rho1 <- 0.003
xihat1_log <- 9.852738 # ML estimate
alphahat1_log <- 33.54351 # MLE
betahat1_e <- 38509614
alphahat1_e <- 4.636787

U1 <- matrix(runif(n1*mb),n1,mb)

alphahatstar1_gamma <- 1:mb*0
xihatstar1_gamma <- 1:mb*0
betahatstar1_e <- 1:mb*0
alphahatstar1_e <- 1:mb*0

for (i in 1:mb)
{
  n1_c=sum(U1[,i] > rho1)
  Zstar1_c_log=xihat1_log*rgamma(n1_c,alphahat1_log)/alphahat1_log
  xihatstar1_gamma[i]=mean(Zstar1_c_log)
  optimall_star=optimize(llminusLGamma,c(1,1e20)*xihatstar1_gamma[i]/sd(Zstar1_c_log),z=Zstar1_c_log)
  alphahatstar1_gamma[i]=optimall_star[[1]]
  Zstar1_e=betahat1_e*(runif(n1-n1_c)**(-1/alphahat1_e)-1)
  optimalstar1_e=optimize(llminusPareto, c(1,1e12),z=Zstar1_e)
  betahatstar1_e[i]=optimalstar1_e[[1]]
  alphahatstar1_e[i]=1/mean(log(1+Zstar1_e/betahatstar1_e[i]))
}

#Confidence intervals for the parameters:
xihatstar1_gamma_CI <- c(quantile(xihatstar1_gamma,0.025),quantile(xihatstar1_gamma, 0.975))
#(9.837348 9.868828)

alphahatstar1_gamma_CI <- c(quantile(alphahatstar1_gamma,0.025),quantile(alphahatstar1_gamma,0.975))
#(33.14042 33.94620)

alphahatstar1_e_CI <- c(quantile(alphahatstar1_e,0.025),quantile(alphahatstar1_e,0.975))
#(2.529495 93210.694254)

betahatstar1_e_CI <- c(quantile(betahatstar1_e,0.025),quantile(betahatstar1_e,0.975))
#(17384278 999987492939)

#
#MC-estimate for the total loss
#
Nhat1 <- rnegbin(m,mu=J*x11,theta=J+alpha1)

```

```

Xhat1 <- 1:m*0

for (i in 1:m)
{
  U=runif(Nhat1[i])
  n_c=sum(U>rh01)
  n_e=sum(U<=rh01)
  Z_c=exp(xihat1_log*rgamma(n_c,alphahat1_log)/alphahat1_log)
  Z_e=b1+betahat1_e*(runif(n_e)**(-1/alphahat1_e)-1)
  Xhat1[i]=sum(Z_c)+sum(Z_e)
}

#
#Nested bootstrap Monte Carlo
#
m1 <- m*mb
Xhatstar1 <- 1:m1*0
ind <- rep(1:mb,m) #replicate 1 to mb for m times
epsilon <- 0.01

Nhatstar1 <- rnegbin(m1,mu=J*xihatstar1_NB,theta=J*alphahatstar1_NB)

for (j in 1:m1)
{
  U1=runif(Nhatstar1[j])
  n1_c=sum(U1>rh01)
  n1_e=sum(U1<=rh01)
  Z1_c=exp(xihatstar1_gamma[ind[j]]*rgamma(n1_c,alphahatstar1_gamma[ind[j]])/alphahatstar1_gamma[ind[j]])
  Z1_e=b1+betahatstar1_e[ind[j]]*(runif(n1_e)**(-1/alphahatstar1_e[ind[j]])-1)
  Xhatstar1[j]=sum(Z1_c)+sum(Z1_e)
}
Xhatstar1 <- matrix(Xhatstar1,byrow=T,m)

#####
###Type 2 and type 3###
#####
#
#Bootstrap estimates of NB:
#
alpha23 <- 0.4686309
xi23 <- 1.3236635

NB23 <- matrix(rnegbin(n*mb,mu=xi23,theta=alpha23),n,mb)

xihatstar23_NB <- 1:mb*0
alphahatstar23_NB <- 1:mb*0

for (i in 1:mb)
{
  xihatstar23_NB[i]=mean(NB23[,i])
  optimal23_star=optimize(llminusNBin,c(0.0000001,100),n=NB23[,i])$min
  alphahatstar23_NB[i]=optimal23_star
}

#
#Bootstrap of Gamma on log scale for the central part and Pareto for the extreme part.
#
n23 <- length(new_fast_s_tp23)
b23 <- 15108527
rho23 <- 0.002

```

```

xihat23_log <- 10.26194
alphahat23_log <- 28.81373
betahat23_e <- 34391349
alphahat23_e <- 2.446868

U23 <- matrix(runif(n23*mb),n23,mb)

alphahatstar23_gamma <- 1:mb*0
xihatstar23_gamma <- 1:mb*0
betahatstar23_e <- 1:mb*0
alphahatstar23_e <- 1:mb*0

for (i in 1:mb)
{
  n23_c=sum(U23[,i] > rho23)
  Zstar23_c_log=xihat23_log*rgamma(n23_c,alphahat23_log)/alphahat23_log
  xihatstar23_gamma[i]=mean(Zstar23_c_log)
  optimal23_star=optimize(llminusLGamma,c(1,1e20)*xihatstar23_gamma[i]/sd(Zstar23_c_log),z=Zstar23_c_log)
  alphahatstar23_gamma[i]=optimal23_star[[1]]
  Zstar23_e=betahat23_e*(runif(n23-n23_c)**(-1/alphahat23_e)-1)
  optimalstar23_e=optimize(llminusPareto,c(1,1e12),z=Zstar23_e)
  betahatstar23_e[i]=optimalstar23_e[[1]]
  alphahatstar23_e[i]=1/mean(log(1+Zstar23_e/betahatstar23_e[i]))
}

#Confidence intervals for the parameters
xihatstar23_gamma_CI <- c(quantile(xihatstar23_gamma,0.025),quantile(xihatstar23_gamma,0.975))
#(10.23328 10.29375)

alphahatstar23_gamma_CI <- c(quantile(alphahatstar23_gamma,0.025),quantile(alphahatstar23_gamma,0.975))
#(28.20578 29.48510)

alphahatstar23_e_CI <- c(quantile(alphahatstar23_e,0.025),quantile(alphahatstar23_e,0.975))
#(1.117826 66996.247717)

betahatstar23_e_CI <- c(quantile(betahatstar23_e,0.025),quantile(betahatstar23_e,0.975))
#(9491445 999999486470)

#
#MC-estimate for the total loss
#
Nhat23 <- rnegbin(m,mu=J*xihat23,theta=J*alpha23)
Xhat23 <- 1:m*0
for (i in 1:m)
{
  U=runif(Nhat23[i])
  n_c=sum(U>rho23)
  n_e=sum(U<=rho23)
  Z_c=exp(xihat23_log*rgamma(n_c,alphahat23_log)/alphahat23_log)
  Z_e=b23+betahat23_e*(runif(n_e)**(-1/alphahat23_e)-1)
  Xhat23[i]=sum(Z_c)+sum(Z_e)
}

#
#Nested algorithm
#
m1 <- m*mb
Xhatstar23 <- 1:m1*0
ind <- rep(1:mb,m) #replicate 1 to mb for m times
epsilon <- 0.01

Nhatstar23 <- rnegbin(m1,mu=J*xihatstar23_NB,theta=J*alphahatstar23_NB)

for (j in 1:m1)

```

```

{
  U23=runif(Nhatstar23[j])
  n23_c=sum(U23>rho23)
  n23_e=sum(U23<=rho23)
  Z23_c=exp(xihatstar23_gamma[ind[j]]*rgamma(n23_c,alphahatstar23_gamma[ind[j]])/alphahatstar23_gamma[ind[j]])
  Z23_e=b23+betahatstar23_e[ind[j]]*(runif(n23_e)**(-1/alphahatstar23_e[ind[j]])-1)
  Xhatstar23[j]=sum(Z23_c)+sum(Z23_e)
}
Xhatstar23 <- matrix(Xhatstar23,byrow=T,m)

#####
###Type 4, type 5 and type9###
#####
#
#Bootstrap estimates of NB
#
alpha459 <- 0.2812828 # MLE
xi459 <- 0.4125248 # MLE

NB459 <- matrix(rnegbin(n*mb,mu=xi459,theta=alpha459),n,mb)

xihatstar459_NB <- 1:mb*0
alphahatstar459_NB <- 1:mb*0

for (i in 1:mb)
{
  xihatstar459_NB[i]=mean(NB459[,i])
  optimal459_star=optimize(llminusNBin,c(0.0000001,100),n=NB459[,i])$min
  alphahatstar459_NB[i]=optimal459_star
}

#Sum up the bootstrap estimates from the separate models
xihatstarNB_sum <- xihatstar1_NB+xihatstar23_NB+xihatstar459_NB
alphahatstarNB_sum <- alphahatstar1_NB+alphahatstar23_NB+alphahatstar459_NB

#Confidence interval for the bootstrap estimates
xihatstarNB_sum_CI <- c(quantile(xihatstarNB_sum,0.025),quantile(xihatstarNB_sum,0.975))
#(5.707675 5.870024)

alphahatstarNB_sum_CI <- c(quantile(alphahatstarNB_sum,0.025),quantile(alphahatstarNB_sum,0.975))
#(1.771981 1.855504)

#
#Bootstrap of Gamma on log scale for the central part and Pareto for the extreme part.
#
n459 <- length(new_fasts_tp459)
b459 <- 8876982
rho459 <- 0.002

xihat459_log <- 9.933211 # MLE
alphahat459_log <- 25.24475 # MLE
betahat459_e <- 999990662494 # MLE
alphahat459_e <- 42783.79 # MLE

U459 <- matrix(runif(n459*mb),n459,mb)

alphahatstar459_gamma <- 1:mb*0
xihatstar459_gamma <- 1:mb*0
betahatstar459_e <- 1:mb*0
alphahatstar459_e <- 1:mb*0

for (i in 1:mb)

```



```

{
  n459_c=sum(U459[,i] > rho459)
  Zstar459_c_log=xihat459_log*rgamma(n459_c,alphahat459_log)/alphahat459_log
  xihatstar459_gamma[i]=mean(Zstar459_c_log)
  optimal459_star=optimize(llminusLGamma,c(1,1e20)*xihatstar459_gamma[i]/sd(Zstar459_c_log),z=Zstar459_c_
    log)
  alphahatstar459_gamma[i]=optimal459_star[[1]]
  Zstar459_e=betahat459_e*(runif(n459-n459_c)**(-1/alphahat459_e)-1)
  optimalstar459_e=optimize(llminusPareto,c(1,1e12),z=Zstar459_e)
  betahatstar459_e[i]=optimalstar459_e[[1]]
  alphahatstar459_e[i]=1/mean(log(1+Zstar459_e/betahatstar459_e[i]))
}

#Confidence intervals for the parameters
xihatstar459_gamma_CI <- c(quantile(xihatstar459_gamma,0.025),quantile(xihatstar459_gamma,0.975))
#(9.873799 9.996240)

alphahatstar459_gamma_CI <- c(quantile(alphahatstar459_gamma,0.025),quantile(alphahatstar459_gamma,0.975))
#(24.19066 26.41963)

alphahatstar459_e_CI <- c(quantile(alphahatstar459_e,0.025),quantile(alphahatstar459_e,0.975))
#(1.49066 117445.41127 )

betahatstar459_e_CI <- c(quantile(betahatstar459_e,0.025),quantile(betahatstar459_e,0.975))
#(13392616 999999936547 )

#
#MC-estimates for the total loss
#
Nhat459 <- rnegbin(m,mu=J*x_i459,theta=J*alpha459)
Xhat459 <- 1:m*0
for (i in 1:m)
{
  U=runif(Nhat459[i])
  n_c=sum(U>rho459)
  n_e=sum(U<=rho459)
  Z_c=exp(xihat459_log*rgamma(n_c,alphahat459_log)/alphahat459_log)
  Z_e=b459+betahat459_e*(runif(n_e)**(-1/alphahat459_e)-1)
  Xhat459[i]=sum(Z_c)+sum(Z_e)
}
#Sum up the total loss from the separate models
Xhat_tot <- Xhat1+Xhat23+Xhat459

#MC-estimate for the reserve based on the separate models
qhat_tot <- sort(Xhat_tot)[(1-epsilon)*m]
#1357399177

#
#Nested algorithm
#
Xhatstar459 <- 1:m1*0
Nhatstar459 <- rnegbin(m1,mu=J*xihatstar459_NB,theta=J*alphahatstar459_NB)

for (j in 1:m1)
{
  U459=runif(Nhatstar459[j])
  n459_c=sum(U459>rho459)
  n459_e=sum(U459<=rho459)
  Z459_c=exp(xihatstar459_gamma[ind[j]]*rgamma(n459_c,alphahatstar459_gamma[ind[j]])/alphahatstar459_gamma[
    ind[j]])
  Z459_e=b459+betahatstar459_e[ind[j]]*(runif(n459_e)**(-1/alphahatstar459_e[ind[j]))-1)
  Xhatstar459[j]=sum(Z459_c)+sum(Z459_e)
}
Xhatstar459 <- matrix(Xhatstar459,byrow=T,m)

```

```
#Sum up the bootstrap MC estimates of the total loss from the separate models and #calculate the
corresponding reserve
Xhatstar_tot <- Xhatstar1+Xhatstar23+Xhatstar459
qhat_star_tot <- apply(Xhatstar_tot,2,sort)[m*(1-epsilon),]

#Confidence interval of the bootstrap MC estimates of the reserve based on a #combination of the separate
models
qhatstar_tot_CI <- c(quantile(qhat_star_tot, 0.025),quantile(qhat_star_tot, 0.975))
#(1262036311 2044171444)

#Standard deviation of the bootstrap MC estimates of the reserv based on a #combination of the separate
models
qhatstar_tot_sd <- sd(qhat_star_tot)
#248308341
```