DIPLOMARBEIT

Quantifying Operational Risk

Documentation of an AMA model

Ausgeführt am Institut für
Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

UNTER DER ANLEITUNG VON

Ao.Univ-Prof. DI Dr.tech. Grill

DURCH

Christian Sagmeister

Aichholzgasse 18, 1120 Wien

Datum

Unterschrift (Student)

Contents

List of Abbreviations

AD test Anderson-Darling test

AMA Advanced Measurement Approach

BEICF Factors reflecting the business environment and internal control systems

BIS Bank of International Settlement

cdf Cumulative Distribution Function

CvM Cramér-von Mises (criterion)

ECDF Empirical Cumulative Distribution Function

EDF Empirical Distribution Function

FFT Fast Fourier transform

GoF Goodness of Fit

KRI Key Risk Indicators

KS test Kolmogorov-Smirnov test

LDA Loss Distribution Approach

MAP Maximum a posterior

MC Monte Carlo

MCMC Markov Chain Monte Carlo

MLE Maximum Likelihood Estimator

MMSE Minimum Mean Square Error

MoM Methods of Moments

OpVaR Operational Value at Risk

ORC Operational Risk Cells

ORM Operational Risk Management

ORX Operational Riskdata eXchange

Acknowledgements

Foremost, I want to thank Karl Grill for his supervision and constant support. Moreover, I owe my gratitude to my colleague Jasmin Sahbegovic, who helped me finding solutions to seemingly unsolvable problems. I also have to thank my superiors Danilo Mercurio and Stephan Kossmeier. Their immense experience and advice have been highly valuable.

Besides them, I have to thank the following persons for their insightful comments and remarks on modelling with Markov Chain Monte Carlo (MCMC):

- Xiaoping Zhou (State University of New York at Stony Brook), who developed important techniques regarding MCMC in operational risk, and who told me how to specify the approach for a particular problem,
- Sojung Kim (Korea Advanced Institute of Science and Technology), who read through the modelling ideas and helped me refine them
- and Florian Huber (WU Wien) who showed me the power of the `R`-package `coda` for MCMC.

Without them, this thesis would look differently.

Chapter 1

Introduction

Begin at the beginning...

Lewis Carroll, Alice in Wonderland

The Basel Committee defines operational risk as

> *"...the risk of loss resulting from inadequate or failed internal **processes**, **people** and **systems** or from **external events**. This definition includes legal risk, but excludes strategic and reputational risk."* [oBS05]

The Basel II capital accord rates operational risk among one of the three risk types for which a bank has to maintain capital reserves to cover their risk exposure. In contrast to the other two risk types, credit risk and market risk, operational risk includes many different inhomogeneous risk drivers like fraud, natural disasters or human error.

When it comes to operational risk, it is also important to categorize the losses according to the business area where they happened - e.g. in retail banking or investment banking. The high diversity was already recognized by the Basel Committee in their 2002 loss data collection exercise, where they defined 56 different basic **Operational Risk Cells** (ORC) based on 8 different business lines and 7 different event types (see Figure 1.1).

The huge inhomogeneity of operational losses is by far not the only challenge when it comes to operational risk. The risk is mainly driven by a few huge losses. The recent USD 13 billion (about EUR 9.5 billion) mortgage settlement of JP Morgan is one example of what size a single loss could possibly have. The potential that one single huge loss could even mean that the bank goes out of business shows the importance of a proper **Operational Risk Management** (ORM) .

The Basel Committee incentivized a sound ORM by the introduction of the **Advanced Measurement Approach** (AMA) for the calculation of the operational risk capital. After meeting several qualitative and quantitative standards (see [oBS11] or [NG06]) for the ORM, the bank has the potential to show to the regulator that it needs to maintain less capital for operational risk than it would have needed with the standard calculation approach.

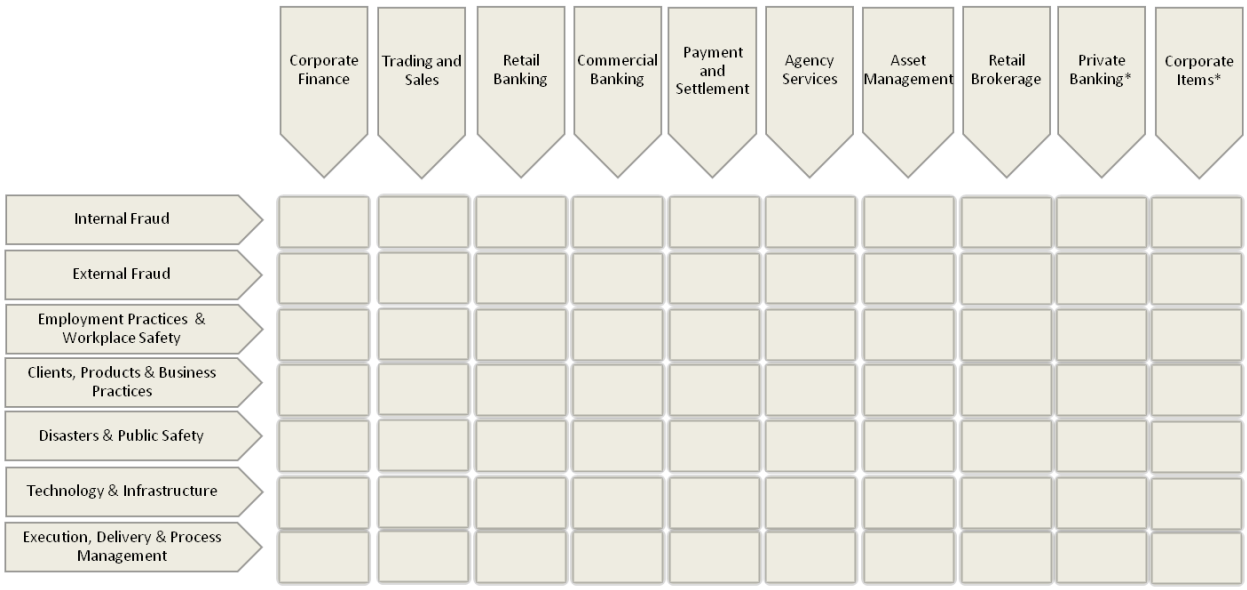

Figure 1.1: *The ORC - matrix according to the Basel Committee ([oBS03]). Business Lines marked with "*" were not included in the original Basel II business lines, but are nevertheless used by the majority of the AMA banks.*

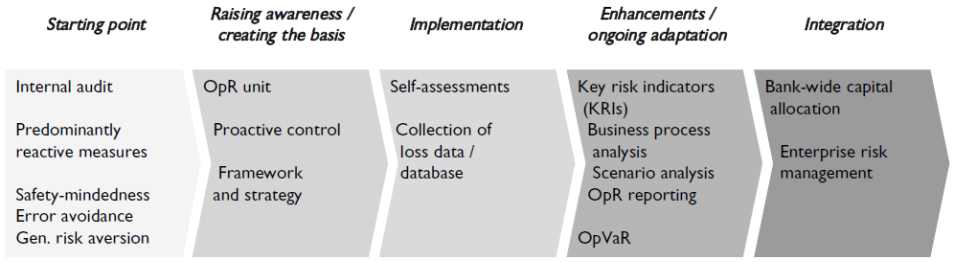

Figure 1.2: *Step-by-Step Introduction of Operational Risk Management for the AMA [NG06]*

To achieve this goal, the bank has to show that its measurement method is capable of capturing the mentioned huge tail events and has to achieve a soundness standard comparable to a 99.9% confidence interval of a one year period ([oBS11]). This measure is called the **Operational Value at Risk** (OpVaR).

According to the Basel Committee, the operational risk measurement system must include the following **four key elements** to meet the soundness standard ([NG06]):

(i) **Internal loss data** are the losses collected within a credit institution. Their systematic collection forms the basis for the quantitative model. The model depends strongly on the quality of loss data, therefore an application for the AMA can only be successful if the bank can show a sufficient data quality process. Moreover, the bank needs to have a data history of at least five years. Nevertheless even a data history of more than five years can never be

sufficient to estimate quantiles of high order. For that reason, external loss data and scenarios are needed.

(ii) **External loss data** are operational losses experienced by other banks. A credit institution has many possibilities to obtain external data: either by buying from a commercial provider or by joining a operational loss data consortium. Often it is preferable to join a consortium because commercial provider often only offer access to publicly known operational losses. The drawback of using consortium data is that it often lacks a proper description of the case because it is made anonymous to protect the loss owner. One example for a data consortium is the Operational Riskdata eXchange (ORX) Association.

(iii) **Scenario analyses** are assessments of operational risk made by experts considering possible situations where operational losses could occur. The advantage of using scenario analyses is a forward-looking aspect, while data is always backward-looking. However, it is often questionable how reliable quantitative estimates by experts are, as they are subject to many biases.

(iv) **Factors reflecting the business environment and internal control systems (BEICF)** indicate, similar to scenario analyses, areas with a high risk of future operational losses *before* they happened. BEICF work on a more quantitative basis than expert estimates. A prominent BEICF are the so-called Key Risk Indicators (KRIs). The credit institution defines on its own triggers for an early warning system, thinking about the roots of possible operational losses.

Examples of KRIs are: (see [NG06])

- staff fluctuation rate
- days of sickness leave
- hours of overtime
- number and duration of system failures
- internal audit findings
- frequency of complaints
- wrong account entries

The thesis will be structured as follows:

Approaches in Operational Risk Modelling

The AMA gives banks a high flexibility in their model approaches. This led to a very diversified universe of different AMA modelling techniques. The Bank of International Settlement (BIS) undertook a survey amongst all AMA banks in 2009 to discover the range of practice for the AMA. Together with these results, which can be found in [oBS09], this chapter will introduce and discuss the most widely followed approaches in operational risk modelling and the mathematical models behind them.

The Model in Practice of the Validation of the Model

The challenge in the AMA is to not only choose between one of the many different modelling methods, but also to adapt them to the unique operational risk profile of a bank. The chapter will explain the difficult procedure of finding the right approaches and how to specify them. Furthermore it will introduce a new technique based on Bayesian inference to include the different AMA key elements.

Furthermore the regulator requires a bank to thoroughly validate the underlying model assumptions. The validation approach will be described in a separate chapter.

The described model is currently being used by a major Austrian bank for their AMA capital calculation.

Appendix

In the course of the modelling work with the Austrian bank a `R`-Package has been developed to aggregate all the functions and techniques that have been designed to find the AMA capital requirement. Because this thesis should focus on the practical implementation of the AMA, the most important codes and their documentation will be provided in the appendix.

Chapter 2

Approaches in Operational Risk Modelling

> All models are wrong, some models are useful.
>
> G. E. P. Box, Robustness in the Strategy of Scientific Model Building

Before it is possible to define a sound risk measure for the AMA, one needs to define a model for quantifying the operational risk of a bank. The industry standard (also mentioned in the guidelines [oBS11], but nevertheless not obligatory) is the so-called **Loss Distribution Approach** (LDA). Instead of looking at the cumulated one-year operational loss, one models the one-year frequency of losses and the loss size (*severity of loss*) independently and puts them together in a second step. This approach, very common in non-life insurance mathematics (see e.g. [Mik09] or [Str97]), has the advantage to make the most out of the available data, as it uses single loss data instead of cumulated yearly loss sums. Moreover, it is well understood by ORM experts (who usually also think in terms of frequency and severity). The disadvantage is that there is no closed formula for the yearly loss distribution, which has to be derived by either Monte Carlo simulations or numerical approximations.

2.1 The Model

The first step is to define reasonable ORC based on the Business Line / Event Type description provided by [oBS03]. If there is not enough data to model each cell of the ORC - matrix, it is also possible to merge cells (e.g. instead of all 56 cells one could just rely on the 7 cells of the event type).
It is assumed that within an ORC all losses occur independently from each other. Moreover we assume that these single losses (denoted by X_i) descend from the same distribution $F(.)$ on $\mathbb{R}_{>0}$ (e.g. log normal).

The (random) yearly frequency is denoted by N. We assume that N is discretely distributed on $\mathbb{N} \cup \{0\}$ (with $P(N = n) =: p_n$) and that the losses X_i occur independently from N. In this framework, the yearly loss S in a particular ORC is now given by

$$S = \sum_{i=1}^{N} X_i.$$

As shown by Abraham Wald in [Wal44] and others, the above assumptions allow for simple formulas for the first moments like

$$\mathbb{E}[S] = \mathbb{E}[X] \cdot \mathbb{E}[N],$$

i.e. the average of the yearly loss is given by a simple multiplication of the average of the single losses and the average frequency. However, we are primarily interested in the calculation of the OpVaR. As mentioned, the OpVaR should achieve a standard comparable to a 99.9% confidence interval, i.e. the OpVaR number for a single ORC should be greater or equal than S with 99.9% probability. Because a bank is also interested in allocating the capital most efficiently, it will allocate the lowest possible number that fulfils this standard, i.e.

$$\text{OpVaR} = \inf\{q \in \mathbb{R} : P(S \leq q) \geq 0.999\}. \tag{2.1}$$

The risk measure depends on $P(S \leq q)$, for which, unfortunately, there is no closed formula like Wald's equation for the first moment. We will therefore heavily rely on numerical methods (see Section 2.4).
Moreover, so far we have only looked at one single ORC. There are many methods to achieve an OpVaR number for the whole operational risk of a bank. The easiest is just summing up the individual OpVaR quantities of the single ORC (mentioned as the most conservative approach in [oBS11]), however the more sophisticated methods will estimate a dependence structure between the ORC.
However, in the beginning we have to concentrate on the two main building blocks: the frequency and the severity.

2.2 Frequency Distribution

In theory, there is no reason to narrow down the range of possible discrete frequency distributions. However, in practice, the two almost exclusively used frequency distributions are the Poisson distribution and the Negative Binomial distribution. A survey of the Bank for International Settlement (see [oBS09]) showed that 93% of the banks used the Poisson distribution as one of their possible frequency distributions, whereas 19% are also using the Negative Binomial distribution as a possible choice.

2.2.1 The Poisson Distribution

The Poisson distribution has many nice properties, one of them is that it just depends on one single parameter. The discrete probabilities are given by

$$p_n = \frac{\lambda^n e^{-\lambda}}{n!}, \quad n = 0, ...$$

for $\lambda > 0$.

Moments: We have very convenient formulas for the first two central moments:

$$\mathbb{E}[N] = \text{Var[N]} = \lambda.$$

Maximum Likelihood Estimator: Another nice property is the simple form of the Maximum Likelihood Estimator (MLE). It equals to the average frequency in the sample:

$$\hat{\lambda} = \frac{\text{\# data points in ORC}}{\text{\# years in data set}}.$$

2.2.2 The Negative Binomial Distribution

The Negative Binomial distribution is not as convenient as the Poisson distribution, nevertheless it can be very useful as it allows for more flexibility in fitting the distribution to the data.

$$p_n = \binom{k+r-1}{n}(1-p)^r p^k, \quad n = 0, ...$$

for some $r > 0$ and $p \in (0, 1)$.

Moments: The formulas for the first two moments are given by

$$\mathbb{E}[N] = \frac{pr}{1-p}, \quad \text{Var[N]} = \frac{\text{pr}}{(1-\text{p})^2}.$$

Maximum Likelihood Estimator: Unfortunately, there is no closed formula for the MLE in the case of a Negative Binomial distribution and we have to stick to numerical optimization methods. However, due to the lack of a sufficient data time frame, it might not be recommendable to use the yearly sample frequencies to estimate the parameters. Therefore, in practice, the following estimation method is used:

(i) Fit a Negative Binomial distribution to monthly (or quarterly) frequency data and receive the parameters $r^{\text{mon}}, p^{\text{mon}}$ (or $r^{\text{qu}}, p^{\text{qu}}$).

(ii) Due to the convolution property of the Negative Binomial distribution the MLE is given by

$$\hat{r} = 12 \cdot r^{\text{mon}}, \hat{p} = p^{\text{mon}}$$

(or $\hat{r} = 5 \cdot r^{\text{qu}}, \hat{p} = p^{\text{qu}}$ for a quarterly estimation period).

The convolution property is only valid if the underlying data set is independently distributed. Therefore it is also reasonable to choose the estimation period in a way that independence over time is still assured.

2.2.3 Choice of Frequency Distribution

In practice, the starting hypothesis is that the frequency distribution is the Poisson (because of the very convenient properties mentioned above). Only if it is reasonable to reject this hypothesis, the Negative Binomial distribution will be assumed instead.

Of course, one could rely on statistical tests like the Kolmogorov-Smirnov test to test the Poisson hypothesis, nevertheless a simple heuristic is often helpful. As mentioned above, for the Poisson distribution we have that the average equals variance. However, in real life data, we often tend to see $\text{Var}[N] > \mathbb{E}[N]$, i.e. **overdispersion**. In this case, it can be reasonable to use a Negative Binomial distribution instead (which fulfils $\text{Var}[N] > \mathbb{E}[N]$).

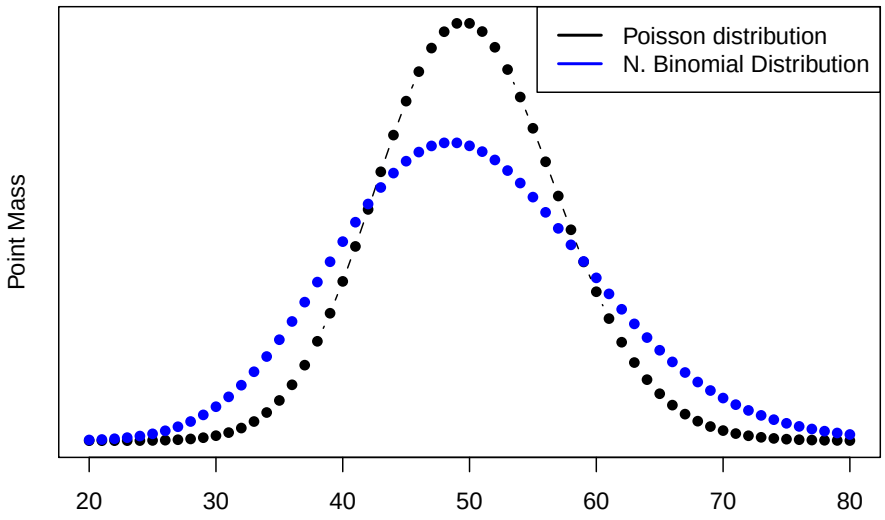

Figure 2.1: *Point probabilities of a Poisson and a Negative Binomial distribution with the same average frequency = 50. The figure shows that the Negative Binomial distribution exhibits a much larger variation than the Poisson.*

2.3 Severity Distribution

2.3.1 Truncated Data Sample

We usually have to fit the severity distribution to a **truncated data sample**, i.e. instead of all the losses we are just able to see losses that are above a given threshold T:

- T as **data collection threshold**:
  Usually, not every operational loss is actually recorded in a data base. That is because it needs time to enter all details (like an exact description of what happened, where it happened, exact

loss amount etc.) which is an effort that simply does not pay off if the loss amount is actually very small. For this reason, banks do not oblige their employees to enter an operational loss if the amount is below a certain threshold. In the case of the ORX database, the data collection threshold amounts to EUR 20,000.

- T as **tail threshold**:
  It seems to be a general property of operational loss data that the losses can hardly be fit with any two-parameter distribution. The consequence could be to take distributions with a high number of parameters, which again leads to numerical issues. [oBS09] shows that only 31% of the AMA banks fit a parametric distribution to the whole range of losses.

  The alternative is quite simple: Instead of trying to fit a complicated distribution to the data, one can concentrate exclusively on the tail that drives the operational risk capital. Usually it is possible to fit a simple two-parameter distribution to tail data, i.e. to losses that are above a predefined tail threshold. Below this threshold, another distribution can be assumed (e.g. empirical).

  In this framework we assume the following cumulative distribution function (cdf) $F(x)$ of the severity (see also [AK06]):

$$F(x) = \begin{cases} F^{\mathrm{b}}(x), & x \leq T, \\ 1 - \left(1 - F^{\mathrm{b}}(T)\right) \cdot \left(1 - F^{\mathrm{t}}(x)\right), & x > T, \end{cases}$$

  where $F^{\mathrm{b}}(x)$ denotes the *body cdf* defined on $[0, \infty)$ and $F^{\mathrm{t}}(x)$ denotes the *tail cdf* defined on $[T, \infty)$. The tail cdf is usually a parametric distribution (according to [oBS09] almost 60% of the AMA banks use either Log Normal, Generalized Pareto, Gamma or Weibull), which needs to be fitted to the tail data - which is again a truncated data sample, as we only look at losses above T.

2.3.2 Truncated vs. Shifted Distribution

Many popular continuous severity distribution families are defined on $(0, \infty)$. However, as seen above we need distributions defined on (T, ∞) for some $T > 0$. The problem therefore is to find a method that fulfils:

$$F_0(.), \text{ cdf on } (0, \infty) \rightarrow F_T(.), \text{ cdf on } (T, \infty)$$

There are two popular methods that are currently being used in the practice, each of them has its pros and cons.

(i) **Truncation:**

$$F_T(x) = \frac{F_0(x) - F_0(T)}{1 - F_0(T)} \cdot \mathbb{1}_{\{x > T\}}$$

Advantages: Assuming that we indeed look at a truncated data sample that originated from $F_0(.)$, we can easily see that

$$P(X \leq x | X \geq T) = \frac{P(T \leq X \leq x)}{P(T \leq X)} \cdot \mathbb{1}_{\{x > T\}} = \frac{F_0(x) - F_0(T)}{1 - F_0(T)} \cdot \mathbb{1}_{\{x > T\}},$$

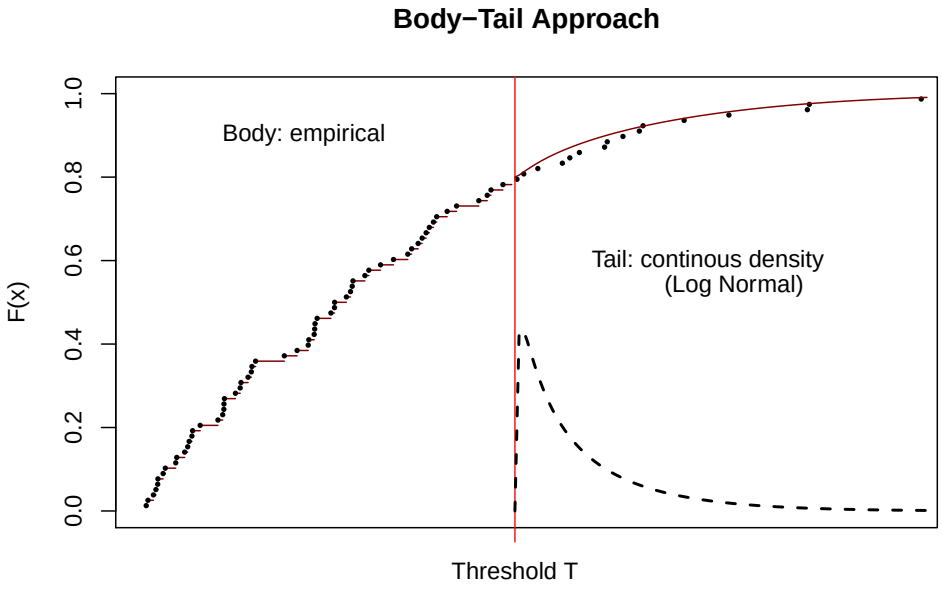

Figure 2.2: *Illustration of the body-and-tail approach*

and therefore the truncation is the natural choice for the distribution. However, we face the problem that we cannot validate that the data sample indeed originated from $F(.)$.

Disadvantages: Numerically, fitting a truncated distribution via maximum likelihood can be very unstable. The problem is particularly severe for large values of $F_0(T)$, i.e. $1 - F_0(T) \approx 0$. This problem has been analysed thoroughly in [Cop11].

(ii) **Shifting:**

$$F_T(x) = F_0(x - T) \cdot \mathbb{1}_{\{x>T\}}$$

Advantages: The MLE in the case of the shifted distribution does in many cases even have an analytical closed form (e.g. for the Log Normal) - and in most of the other cases, it can be obtained by solving a well behaved optimization problem.

Disadvantages: If the probability density function f_0 of $F_0(.)$ fulfils $f_0(0) = 0$ we will have $f_T(T) = 0$. This can lead to numerical problems in the maximum likelihood routine if the data contains losses right above the threshold, where the density equals 0.

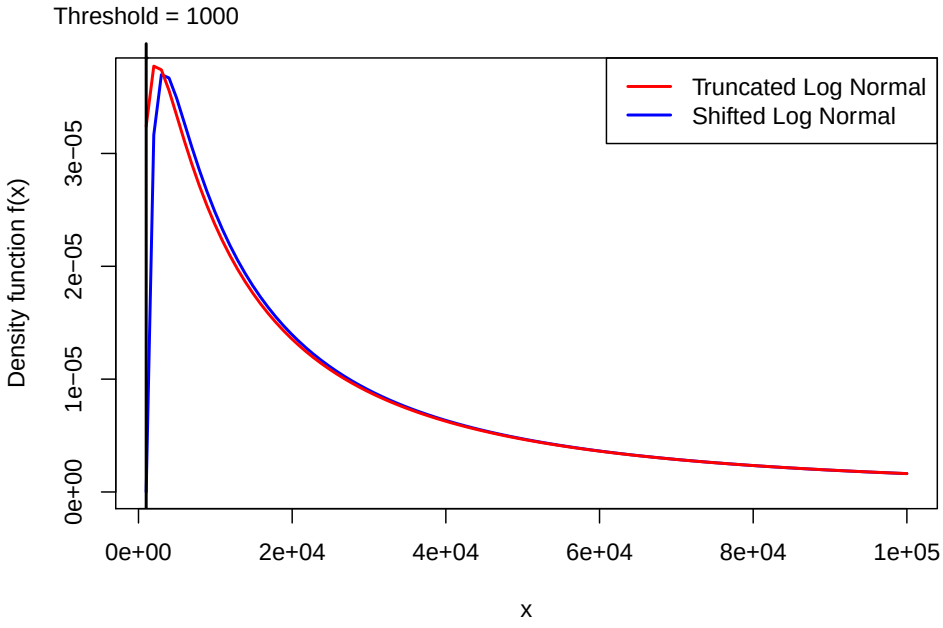

Figure 2.3: *The figure shows the difference between a truncated density and a shifted log normal density with parameters* $\mu = 10$, $\sigma = 1.5$.

2.3.3 Popular Distribution Families

In the spirit of the above discussion of transformation methods only give candidates for distributions on $(0, \infty)$ will be given.

Name	***Probability density function*** $f(x)$
Log Normal	$\frac{1}{x\sigma\sqrt{2\pi}}\, e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}$
Generalized Pareto	$\frac{1}{\sigma}\left(1+\frac{\xi(x-\mu)}{\sigma}\right)^{\left(-\frac{1}{\xi}-1\right)}$
Gamma	$\frac{x^{k-1}e^{-\frac{x}{\theta}}}{\theta^k\Gamma(k)}$
Burr	$ck\frac{x^{c-1}}{(1+x^c)^{k+1}}$
Weibull	$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$

2.3.4 Distribution Fitting Methods

There are several methods to obtain the parameters of a distribution. For each method, we assume that our observations are independent and distributed with a density function $f(.|\theta)$ with parameter θ.

Maximum Likelihood It is one of the best known and most widely followed methods to fit a distribution to a given data sample. The maximum likelihood method means maximizing the

joint density function of the observations (called the likelihood)

$$\ell(\theta \,|\, x_1, \ldots, x_n) := \prod_{i=1}^{n} f(x_i|\theta)$$

with respect to the parameter θ. In other words, the MLE satisfies

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} \ell(\theta \,|\, x_1, \ldots, x_n).$$

For numerical reasons, the likelihood is often replaced by the logarithm of the likelihood function, the *log likelihood*. In the case of a truncated probability distribution with density $f_T(.) = (f_0(.))/(1 - F_0(T))$ the log likelihood takes the form

$$\log \ell(\theta \,|\, x_1, \ldots, x_n) = \sum_{i=1}^{n} \log f_T(x_i|\theta) = \sum_{i=1}^{n} \log f_0(x_i|\theta) - n \log(1 - F_0(T)).$$

The MLE has many nice properties (asymptotic consistency, asymptotic normality...). However, these properties only hold if the data sample does not differ too much from the assumed probability distribution. In operational risk with its huge tail events, it is often useful to use other, robust methods like quantile matching or Bayesian methods.

Methods of Moments (MoM) The idea of the MoM is to match the empirical sample moments (e.g. sample mean, sample variance...) with the theoretical ones by finding the right distribution parameters. As shown as in [Han82] this method can be generalized to minimize the asymptotic variance of the estimator (*generalized methods of moments*).

Quantile Matching Quantile matching is very similar to MoM, however more popular amongst operational risk modellers, because it is possible to give more weight to the tail. Instead of matching moments of the distribution, one matches the theoretical distribution quantiles with the sample moments. For a thorough discussion of the method in an operational risk context have a look at [LR13].

Bayesian Methods In a Bayesian framework the parameter uncertainty is seen as a randomness of the parameter, i.e. one assumes a probability distribution $\pi(\theta)$ on the parameters (the so-called *prior* distribution). With observation, it is possible to update the prior distribution with the new information and receive the *posterior* distribution $\pi(\theta|x)$ of the parameters. This is done via the famous Bayes' theorem

$$\pi(\theta|x) = \frac{\ell(\theta|x)\pi(\theta)}{\int \ell(\vartheta|x)\pi(\vartheta)d\vartheta}. \tag{2.2}$$

From this posterior distribution there are several methods to receive an estimator. The **Maximum a posterior** (MAP) estimate is the mode of the posterior distribution. The mean of the posterior distribution has the property of minimizing the variance of the estimator, it is called the **Minimum Mean Square Error** (MMSE) estimator.

Many different modellers tried to introduce Bayesian methods in Operational Risk ([She11] gives a good overview). However, most of these models offer low flexibility when it comes

to the underlying distribution. The reason is the computation of the posterior distribution. Because of the integral in the denominator of the Bayes' theorem (2.2), explicit expressions of the posterior distribution exist only in special cases. These special cases include the so-called conjugate distributions - in this case the posterior is in the same distribution family as the prior distribution.
This means, to obtain a proper analytical expression of the posterior distribution and therefore an estimator of the parameters, the modeller is required to only choose the prior *as well as* the underlying sample distribution from a very small set of these special cases (and in general it is not possible to choose a truncated distribution).

A possible remedy is the use of numerical methods. The most widely followed numerical Bayes method is **Markov Chain Monte Carlo** (MCMC). Instead of an analytic expression of the prior, this method makes it possible to draw from the posterior distribution. The MCMC method works with the so-called Metropolis-Hastings algorithm:

- Define some starting values θ_0.
- Given θ_{t-1}, draw a sample X from a (multi-dimensional) Normal distribution with mean vector θ_{t-1} and covariance matrix Σ and define $\hat{\theta}_t = \theta_{t-1} + X$.
- With the observations calculate the likelihood ratio

$$LR = \frac{\ell(\theta_t|x) \cdot \pi(\theta_{t-1})}{\ell(\theta_{t-1}|x) \cdot \pi(\theta_{t-1})}.$$

- Define $\alpha = \min(LR, 1)$.
- Generate a uniform random variable U on $(0, 1)$ and set

$$\theta_t = \begin{cases} \hat{\theta}_t, & U < \alpha \\ \theta_{t-1}, & else. \end{cases}$$

Each θ_t is a draw from the posterior distribution. With a sufficient sample size one can now use empirical estimates to derive e.g. the MAP or the MMSE estimator.

The advantage of this method is that it works with almost any kind of prior or likelihood function. The drawback is that it requires a proper analysis of the convergence of the Markov chain $(\theta_t)_t$. Especially in the case of truncated distributions the convergence can be very slow or the chain could even be divergent. [ZGFT13] provided some ideas to ensure convergence in the case of truncated distributions and showed that the method is very stable even for small sizes, providing much more robust results than the MLE.

2.4 Numerical Methods for Determining the OpVaR of a Single ORC

> The purpose of computing is insight, not numbers.
>
> Richard W. Hamming

Once the severity distribution and the frequency distribution has been determined, the compound loss distribution is theoretically already identified. In practice however, one has to rely on numerical methods to receive quantities based on the compound distribution like the OpVaR. The three most common methods are **Monte Carlo** (MC), **Fast Fourier transform** (FFT) and **Panjer Recursion**.

2.4.1 Monte Carlo

The MC method is very simple and intuitive. One draws realizations from the compound loss distribution (see Figure 2.4) and derives the empirical estimates from the sample.

The major drawback of the MC method is the high computational effort. To receive sufficiently exact estimates of the OpVaR one needs very large sample sizes.
To increase the computation speed, [EF09] offers a good overview of two faster alternatives to the MC method: the FFT method and the Panjer Recursion.

2.4.2 Fast Fourier Transform

Even though we do not have a simple expression for the distribution function of the compound loss variable S, we can derive a fairly simple expression for the **characteristic function** of S. The characteristic function $\phi_S(t)$ of S is defined as

$$\phi_S(t) = \mathbb{E}[e^{itS}], \quad t \in \mathbb{R}.$$

Assuming that the frequency is Poisson distributed with parameter λ, we have that (see [Ger79])

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

where ϕ_X is the characteristic function of the severity distribution. Assuming that the frequency is Negative Binomial distributed with parameter r and p, we have

$$\phi_S(t) = \left(\frac{1-p}{1-p\cdot\phi_X(t)}\right)^r.$$

FFT is a numerical algorithm which is available for many different programming environments (e.g. `fft` in R). Given a discrete distribution function on a set with M elements with probability mass p_k, it returns the discrete Fourier transform

$$\hat{\phi}_j = \sum_{k=1}^{M} p_k e^{i2\pi jk/M}, \quad j = 0, 1, \ldots, M-1,$$

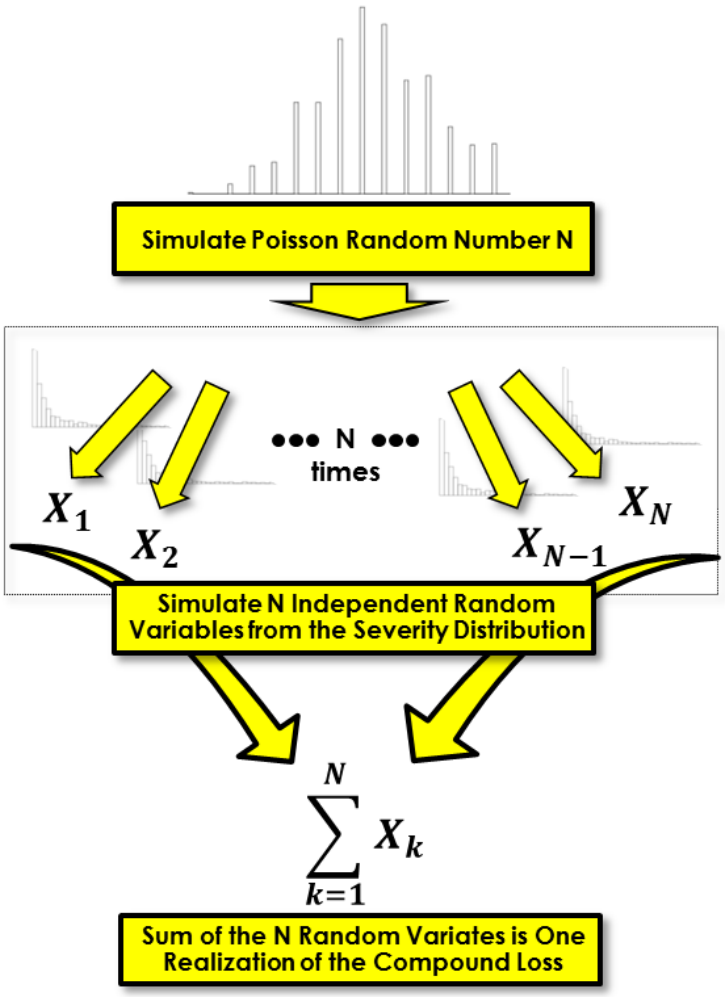

Figure 2.4: *Illustration of the MC method to determine the distribution of the compound yearly loss S.*

which is a discrete approximation of the characteristic function. There is also an inverse FFT (in R: `fft(..., inverse = TRUE)`) which returns a distribution function given a DFT. The FFT method to determine the OpVaR works as follows: (see also [EF09])

(i) Discretize the severity distribution, i.e. transform the (in general continuous) distribution to a discrete distribution on the set $\{0, h, 2h, ..., (M-1)h\}$ for some predefined bandwidth h and truncation point M. This can be done by assigning the probability $P(kh \leq X < (k+1)h)$ to either kh (forward difference) or $(k+1)h$ (backward difference) - or assigning the probability $P(kh - h/2 \leq X < kh + h/2)$ to kh (stochastic rounding).

(ii) Use FFT on the discretized severity distribution to receive an approximation of $\phi_X(.)$. Use one of the above formulas (depending on which frequency distribution was chosen) to receive an approximation of $\phi_S(.)$.

(iii) Use the inverse FFT to finally obtain the distribution of S.

Numerical errors can descend from mainly two sources:

Discretization Error This error can be reduced by using smaller h. Moreover, by using hte forward and backward difference discretization one can get error bounds for the discretization error.

Aliazing Probability mass that lies above the truncation point Mh will erroneously be distributed within the discretization grid. This error is called the aliazing error. In [GH99] an exponential tilting method was presented that can reduce the error significantly: Before applying the FFT on the discretized severity distribution, one multiplies transforms the point masses p_j by applying

$$p_j \to (e^{-\theta j}) \cdot p_j, \quad j = 0, \ldots, M-1,$$

for some $\theta > 0$. Then proceed with step (ii) and (iii) with the transformed distribution. After step (iii), an inverse transformation step (multiplying by $e^{\theta j}$) completes the procedure.

2.4.3 Panjer Recursion

This method is based on a convenient recursion formula for the compound probabilities, that was first described in [Pan81].
Assume that the severity distribution is a discrete distribution on $\mathbb{N}$ with point masses p_k. Then the probability point masses q_n of the compound loss S are given by the recursion

$$q_n = \sum_{j=1}^{n} \left(a + \frac{bj}{n} \right) p_j q_{n-j}.$$

As described for the FFT method, we can use stochastic discretization to receive a discrete severity distribution and make use of the Panjer recursion.
This method gives the exact probability distribution of S (besides the discretization error). However, [EF09] showed that FFT can be considerably faster in computation.

2.5 Dependence Structures

So far, we only discussed models based on one single ORC. Assuming we completed the modelling procedure for each our M ORC (e.g. $M = 56$ if one decides to model on each Basel II cell), we obtain M loss distributions

$$S_j = \sum_{k=1}^{N_j} X_{j,k}, \quad j = 1, \dots, M.$$

To having been able to use the LDA, we used the following independence assumptions so far:

(i) **Independence of the single losses $X_{j,k}, k = 1, \dots$ within each ORC**

(ii) **Independence between $X_{j,k}, k = 1, \dots$ and the frequency N_j**

The question to the operational risk modeller is now *where to assume dependence* between the ORC. According to [oBS09] 21% of the AMA banks assume dependency between the frequencies N_j, 12% assume dependency between the severities $X_{j,k}, j = 1, \dots, M$ (even though this could violate independence assumption (i)) and 33% assume dependency between the compound losses S_j.

After this first choice the next choice that has to be made is *how* to model dependence. [oBS09] showed that the majority of operational risk modellers use a **copula** approach.

A copula is a joint cumulative distribution function of a M-dimensional random vector $(U_1, U_2, ..., U_M)$ on the unit cube $[0,1]^M \to [0,1]$ with uniform marginals:

$$C(u_1, u_2, ..., u_M) = P(U_1 \leq u_1, U_2 \leq u_2, \dots, U_M \leq u_M)$$

The fundamental result regarding copulas is the theorem of Sklar (see [Skl59]). It states that for any joint distribution $F(x_1, x_2, ..., x_M)$ with marginals $F_1(x_1), F_2(x_2), ..., F_M(x_M)$ there exists a Copula C such that

$$F(x_1, x_2, ..., x_M) = C\Big(F_1(x_1), F_2(x_2), \dots, F_M(x_M)\Big).$$

In other words, any joint distribution can be described by its marginals and a copula to describe the dependence structure.

Conversely, a copula composed with random variables is a realization of the joint distribution of those random variables. Therefore, if the dependence is assumed on the compound losses, we can use the following algorithm to simulate realization of the total operational loss:

(i) Model aggregate loss distributions for each ORC

(ii) Fit copula to dependence structure of the ORC

(iii) Simulate a sample $(U_1, U_2, \dots, U_M)$ from the copula distribution

(iv) The final sample is constructed as

$$(S_1, S_2, \dots, S_M) = \big(F_1^{-1}(U_1), F_2^{-1}(U_2), \dots, F_M^{-1}(U_M)\big)$$

(v) Calculate the OpVaR of the (empirical) distribution of the sum of random variables

$$S_1 + S_2 + \cdots + S_M.$$

An open question still remains: which copula is the right one for modelling the operational risk dependence structure?

A simple way to construct families of copulas is to take a parametric family of multi-dimensional distributions $F(.|\theta)$ and define

$$C_{F|\theta}(u_1, u_2, ..., u_M) = F\Big(F_1^{-1}(u_1|\theta), F_2^{-1}(u_2|\theta), \ldots, F_M^{-1}(u_M|\theta)\Big|\theta\Big).$$

Popular examples are the multivariate Normal distribution (Gaussian copula) and the multivariate t distribution (t-copula).
The survey in [oBS09] states that amongst the banks who use copulas, almost all use either Gaussian copula or t-copula. The Gaussian copula is very easy to fit to data (it is only necessary to estimate the correlation), however it does have a major disadvantage when it comes to the dependence between very high tail losses. As shown as in [ELM03] using a Gaussian copula implies asymptotic *tail independence*. Therefore, to introduce a dependence between very high losses in the tail, one could instead rely on the t-copula with tail dependence.

For the t-copula, the parameter θ consists of the elements of the correlation matrix Σ and the degrees-of-freedom parameter df.

The Σ Matrix is estimated by utilizing a robust measure of rank correlation, Kendall's Tau τ_K. There exist the following relationship between τ_K and the elements of Σ (see [MFE10]):

$$\tau_k(x_i, x_j) = \frac{2}{\pi} \arcsin \Sigma_{ij} \tag{2.3}$$

After estimating Σ, one can finally obtain df by using maximum likelihood:

(i) Estimate Marginals $\hat{F}_1(.), \hat{F}_2(.), ..., \hat{F}_M(.)$ (aggregate loss distributions)

(ii) Estimate Copula parameters by MLE:

$$\hat{\theta} = \arg\max_{df} \left(\sum_{t=1}^{T} \log c\Big(\hat{F}_1(x_1), \hat{F}_2(x_2), \ldots, \hat{F}_M(x_M)\Big|df, \Sigma\Big) \right),$$

where c denotes the t-copula density.

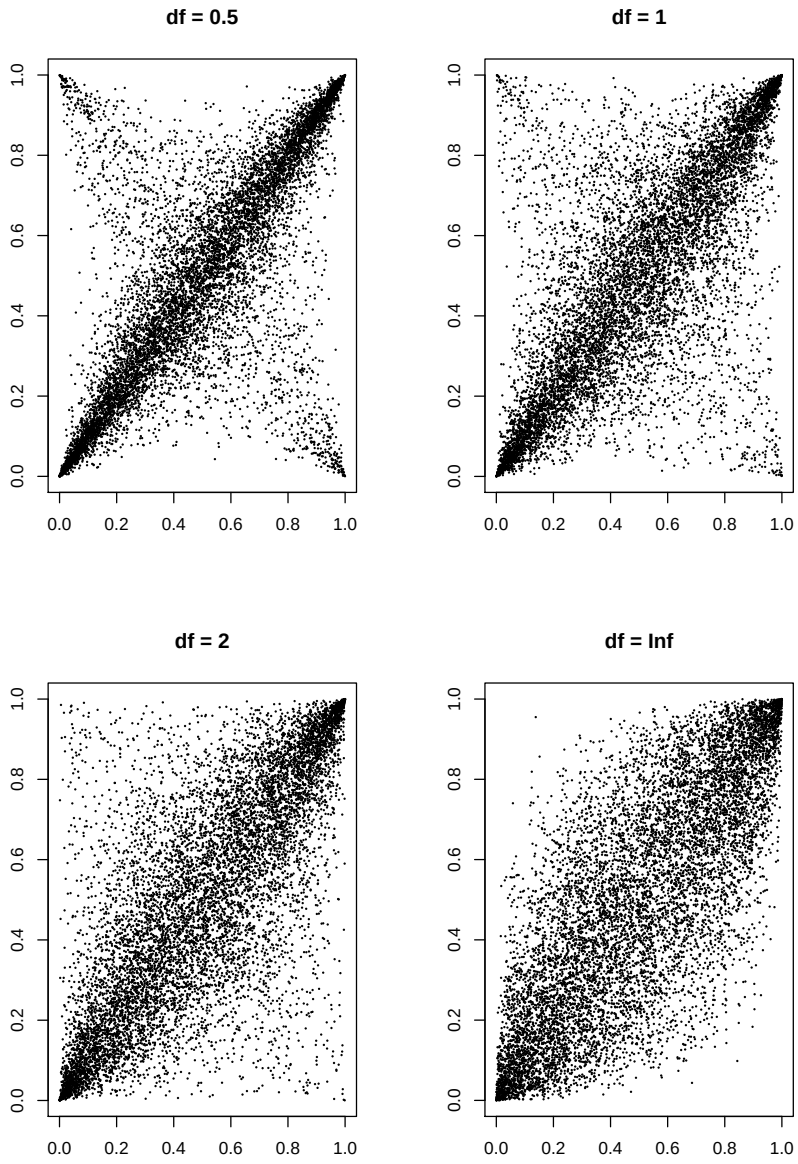

Figure 2.5: *Comparison between simulations of t-copulas with different degrees of freedom (df). $df = \infty$ is the Gaussian copula.*

Chapter 3

The Model in Practice

> In theory, there is no difference between theory and practice. But, in practice, there is.
>
> Jan L. A. van de Snepscheut

As shown in the previous chapter, there is a huge variety of different AMA modelling techniques. The challenge for each bank is to choose the most adequate approach for their specific operational risk profile. Furthermore, to implement the different key elements, the modeller often needs to develop new approaches to achieve the Basel II soundness standards. The chapter will describe these difficulties and how to handle them based on the operational risk profile of a major Austrian bank. Figure 3.1 gives an overview of all the model steps.

Throughout this chapter, the following notation will be used:
Y . . . number of years
n . . . number of observations

3.1 Choice of ORC

With the help of the experts in ORM it was decided to choose the ORC according to the **7 event types**. This has several reasons:

- The amount of loss data is not sufficient to model on each of the single 56 ORC proposed by the Basel committee.
- Using the business lines as ORC instead of the event types has the disadvantage that the losses within a business line are far less homogeneous than losses within an event type.
- For the inclusion of insurance in the model it is also recommendable to model on event types, as insurances usually offer polices based on event types.

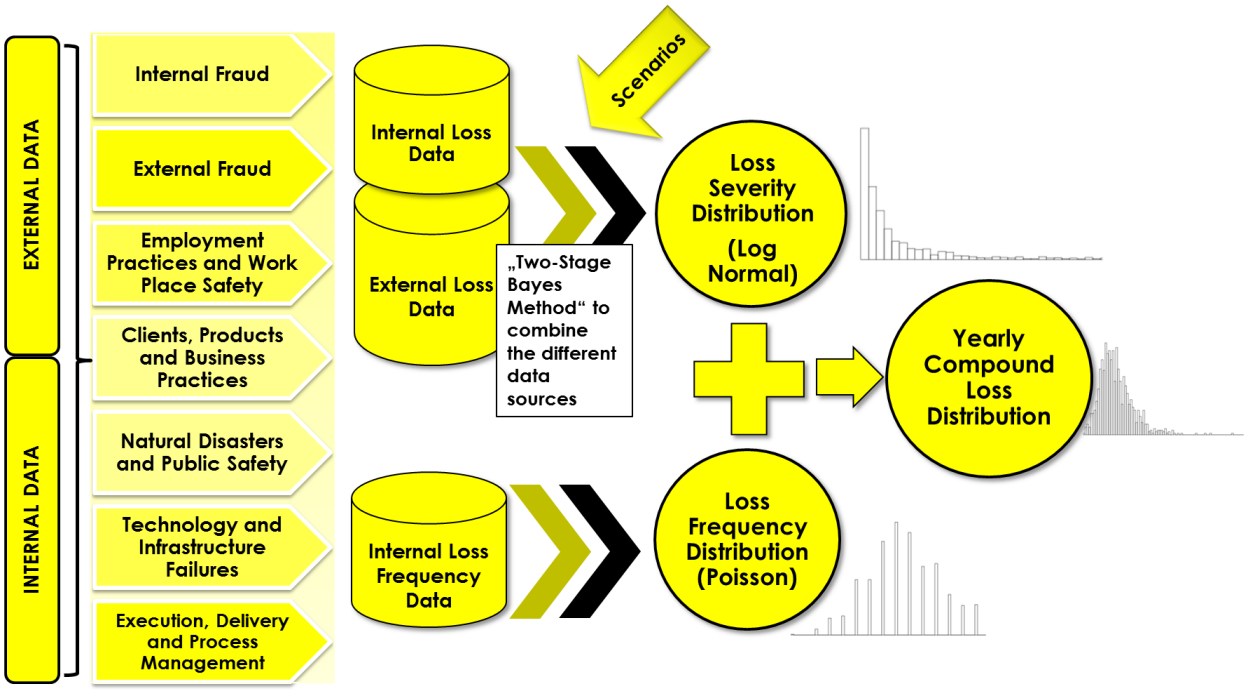

Figure 3.1: *Illustration of the AMA model for a single ORC*

3.2 Frequency Distribution Step

With the methods described in Section 2.2 a frequency distribution is estimated for each event type. Because of its simplicity the Poisson distribution is always the preferred one. However, for some event types the Poisson assumption cannot be validated. The validation process works as follows:

(i) Fit a Poisson distribution to the quarterly loss frequencies N_{qu} of a particular event type

(ii) Calculate a 95% confidence interval by assuming independence of the quarterly loss frequencies. The lower and upper end of the confidence interval (c_ℓ, c_u) are given by the equations

$$0.025 = 1 - P(N_{\text{qu}} > c_\ell)^{4 \cdot Y}, \quad 0.975 = P(N_{\text{qu}} \leq c_u)^{4 \cdot Y}.$$

Note that the event that any of the observed quarterly loss frequency is below c_ℓ is the complementary event that all observed independent events are above c_ℓ. If there are any quarterly frequency observations outside of this confidence interval, a Negative Binomial distribution is estimated instead.

3.3 Severity Distribution Step

3.3.1 Model Assumptions

We assume the following cdf $F(.)$ of the severity:

$$F(x) = \begin{cases} \hat{F}(x), & x \leq T, \\ 1 - \left(1 - \hat{F}(T)\right) \cdot \left(1 - \Phi\left(\frac{\log(x-T)-\mu}{\sigma}\right)\right), & x > T, \end{cases} \tag{3.1}$$

where $\hat{F}(.)$ denotes the empirical cumulative distribution function (ECDF) and $\Phi(.)$ the cdf of the standard normal distribution. The tail threshold $T > 0$, $\mu \in \mathbb{R}$ and $\sigma > 0$ are the parameters to be determined.

Only the shifted log normal is taken as possible tail distribution. This choice was made after considering two points:

- In the course of the modelling procedure not only the distribution parameters μ and σ are fitted to the data, but also the tail threshold T (many banks fix the tail threshold to some arbitrary number instead of fitting it). Therefore we already have enough flexibility by just using one distribution family, the log normal distribution.

- Instead of a truncated distribution a shifted distribution will be used. As mentioned in the previous chapter, shifted distributions offer a higher numerical stability, which will be especially important for our MCMC fitting technique later on.

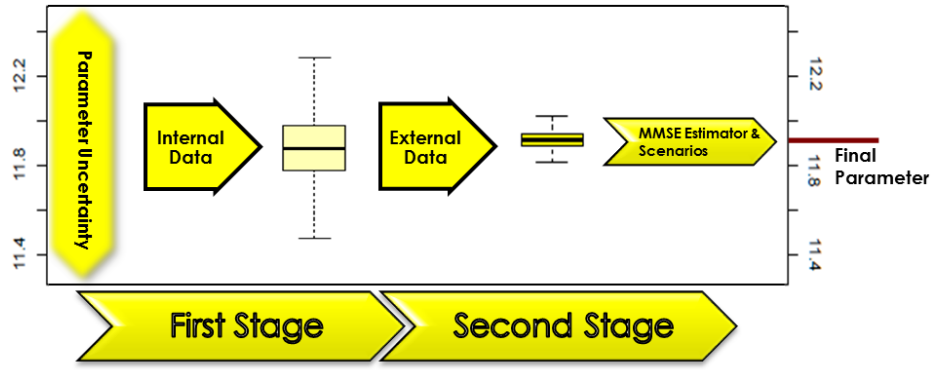

Figure 3.2: *The Two-Stage Bayes approach includes the different AMA elements: internal data, external data and scenarios. Each stage reduces the uncertainty in the parameter estimate.*

3.3.2 The Two-Stage Bayes Approach

> In the first place, the best way to convey to the experimenter what the data tell him about θ is to show him a picture of the posterior distribution.
>
> G.E.P. Box & G.C. Tiao, Bayesian Inference in Statistical Analysis (1973)

To calculate the OpVaR, we extrapolate far beyond our internal data. Therefore we have to expect a large uncertainty in the parameter estimates. With external data we have a possible source of information to decrease this uncertainty. A Bayesian approach is a way to formalize this: We model the uncertainty of the parameters by assuming that they are random and follow some distribution. From our internal data we can derive this distribution empirically. Afterwards we use Bayesian updating with external data to narrow this distribution to a smaller confidence interval and thus reducing the uncertainty (see Figure 3.2).

Stage One

In a first step we assume a non-informative prior, **Jeffrey's prior**, on the parameters μ and σ, i.e. we assume to have no knowledge about the parameters a priori. Jeffrey's prior, introduced in [Jef61], is defined as the square root of the determinant of the Fisher information matrix. It is non-informative in a sense that it assigns uniform weight on the distributions indexed by the parameters (μ, σ) (see [Bal97]). Another notable property of Jeffrey's prior is its translation invariance (it includes no information about the location of the distribution, e.g. it does not depend on μ in the case of the normal distribution). The density of this prior in the case of a normal distribution is given by (see [Roe02])

$$\pi(\mu, \sigma) = \frac{\sqrt{2}}{\sigma^2}.$$

In a first step we want to use our internal data set to determine a distribution of our parameters with less uncertainty. It is not trivial to determine this *posterior distribution* analytically, therefore we decided to use the MCMC method (see Section 2.3.4). In the case of a shifted log normal distribution it works as follows:

- We define some starting values $\theta_0 = (\mu_0, \sigma_0)$.
- Given θ_{t-1}, we draw a sample (X, Y) from a two dimensional Normal distribution with mean vector θ_{t-1} and covariance matrix Σ and define $\hat{\theta}_t = (\mu_{t-1} + X, \sigma_{t-1} + Y)$.
- We calculate the likelihood ratio

$$LR = \frac{\left(\prod_i \varphi(\log(x_i^{int} - T), \mu_t, \sigma_t)\right) \cdot \pi(\mu_t, \sigma_t)}{\left(\prod_i \varphi(\log(x_i^{int} - T), \mu_{t-1}, \sigma_{t-1})\right) \cdot \pi(\mu_{t-1}, \sigma_{t-1})},$$

  where x_i^{int} are all data points of our internal data above the threshold and φ the density function of the normal distribution. For the moment, we assume the tail threshold T as given. The final threshold T will be determined in a later step.
- We define $\alpha = \min(LR, 1)$.
- We generate a uniform random variable U on $(0, 1)$. We set

$$\theta_t = \begin{cases} \hat{\theta}_t, & U < \alpha \\ \theta_{t-1}, & else. \end{cases}$$

Each θ_t is a sample from the posterior distribution of μ and σ.

As suggested in [GRG96], we choose the covariance matrix of the random walk Σ such that the acceptance ratio, i.e. how often $U < \alpha$ in a way that it is at about 25 percent. In particular we will look at Σ of the form

$$\begin{pmatrix} \rho & 0 \\ 0 & \rho/2 \end{pmatrix}$$

with $\rho > 0$ such that the observed acceptance ratio is at the target acceptance ratio of 25 percent.

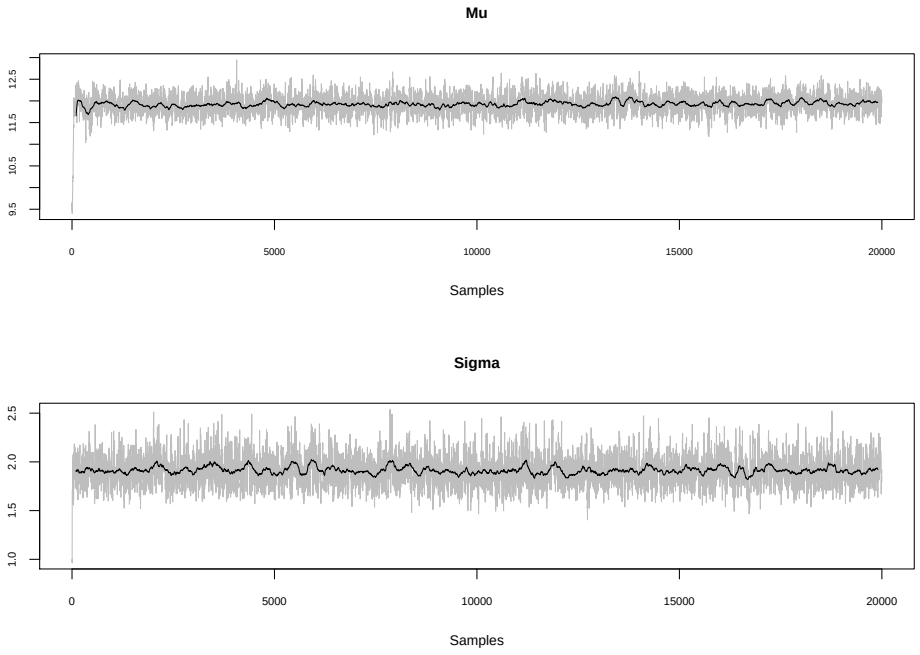

Figure 3.3: *The two plots show the convergence of the θ_t. Even for different starting values θ_0 we have a fast convergence (in a sense that the distribution of θ does not change). Nevertheless we define an initial burnin period of 10 Percent.*

To ensure the choice of the right sample size for the markov chain it is required to do convergence analysis. For that reason, we will use the `R`-package `coda` ([PBCV06]). It provides several tools for our purpose.

First of all, it is worth mentioning that drawing from the Markov chain $(\theta_t)_t$ is not the same as drawing from an independent sample of the posterior distribution. The reason is the positive autocorrelation in the chain, which is defined as

$$R(t) := \frac{\mathrm{E}\left[(\theta_s - \mathrm{E}[\theta_s])(\theta_{s+t} - \mathrm{E}[\theta_s])\right]}{\mathrm{E}\left[(\theta_s - \mathrm{E}[\theta_s])^2\right]}, \quad t > 0,$$

for some predefined *lag* t. The simple rule is: the lower the autocorrelation, the lower the required sample size. `coda` provides the function `effectiveSize()` that calculates the corresponding size for an independent sample, which is usually much lower than for the MCMC. According to experience, one has to at least generate a ten times larger sample to receive the same results as compared to an independent sample.

Besides the test for autocorrelation, there is a variety of statistical tests for convergence analysis. In [G$^+$91] it was proposed to test for the equality of the means of the first and last part of the Markov chain. As shown in [G$^+$91], under the null hypothesis of equal means, the difference of the means divided by its standard error, which is estimated by taking into account any autocorrelation,

is asymptotically normal distributed.
A proposed proportion of the two parts for estimating the means is the first 10% and the last 50%, however it is also recommended to try different proportions as well (see Figure 3.4). `coda` provides the function `geweke.diag()` to get the standardized difference of the means.

Another useful statistical test was proposed in [HW83], which was programmed in `coda` as `heidel.diag()`. The so-called Heidelberg-Welch test checks the null hypothesis that the markov chain sample was generated from a stationary distribution. The idea is that an desirable result would be that the test cannot reject the null even for a small proportion of the chain.

The test is performed for the first 10%, then for the first 20%, and so on, until the first 50%. If the test rejects the null for any of these samples, it would indicate that a larger MCMC sample is needed.

The last test for stationarity is especially important for the next step: fitting a continuous distribution to the markov chain observations, to make things more tractable for the next step.

We assume that μ follows a normal distribution with mean τ and variance ν and for σ we assume that $\frac{1}{\sigma^2}$ follows a Gamma distribution with shape α and rate β ($\sigma \sim f_{\mathcal{G}^{-2}}$). Furthermore we assume the independence of σ and μ.

We fit the parameters via the maximum likelihood estimation.

Finally we can derive the following posterior distribution of the parameters:

$$\pi(\mu, \sigma | x^{int}) = \varphi(\mu|\tau, \nu) \cdot f_{\mathcal{G}^{-2}}(\sigma|\alpha, \beta).$$

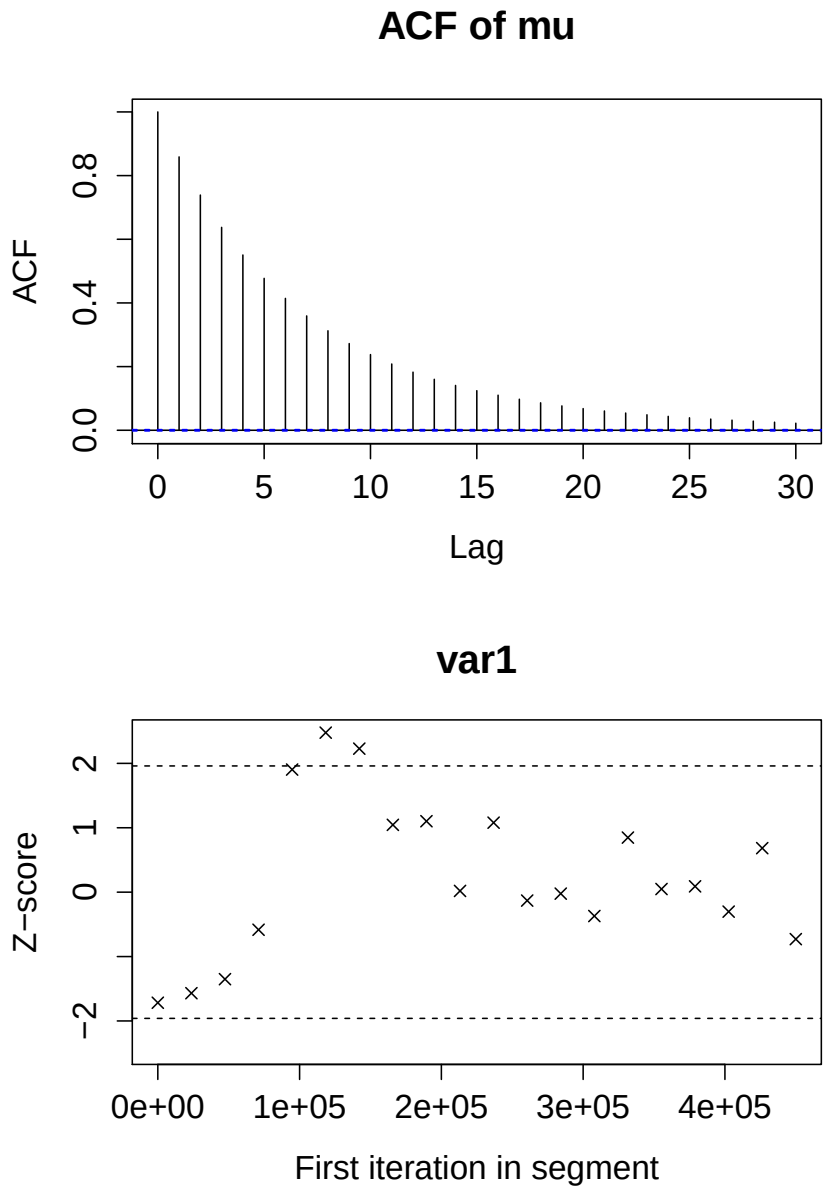

Figure 3.4: *The above plot shows the markov chain autocorrelation function (in the figure above: the markov chain for μ). A fast dropping acf is a sign of good convergence. The second picture shows the Gedeke test statistic for different proportions of the markov chain. The first half of the markov chain is divided into 20 bins, and consequently the test statistic proposed in [G^+ 91] is calculated by excluding the first segement, then also the second segment, then also the third segment etc. The plot also provides confidence intervals ($\alpha = 0.05$) for the test statistics.*

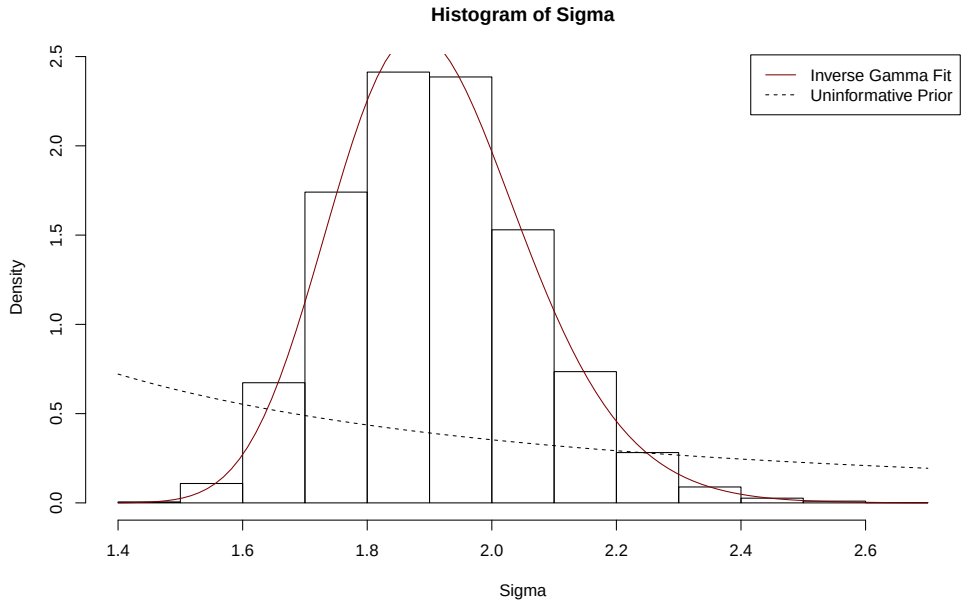

Figure 3.5: *Internal data alone already leads to a great reduction of parameter uncertainty.*

Stage Two

We will now use external data to reduce parameter uncertainty even more. Therefore we repeat the procedure from the step before with the external data instead of internal data, with only three differences.

(i) Instead of Jeffrey's prior we use the posterior distribution given internal data, i.e.

$$\pi^{ext}(\mu, \sigma) = \pi(\mu, \sigma | x^{int}).$$

(ii) As already mentioned ORX data is collected only above a threshold of $T^{\text{ORX}} = 20{,}000$. If $T \geq T^{\text{ORX}}$, we can ignore this fact, because we only look at data above this threshold. But if $T < T^{ORX}$ we have to account for the fact of this *truncated data sample*. This can be done by using the truncated normal density function for determining LR:

$$\varphi^{trunc}\left(\log(x-T), \mu, \sigma\right) = \frac{\varphi\left(\log(x-T), \mu, \sigma\right)}{1 - \Phi\left(\frac{\log(T^{\text{ORX}} - T) - \mu}{\sigma}\right)} \mathbb{1}_{x \geq T^{\text{ORX}}},$$

and we set

$$\varphi^{ext} = \begin{cases} \varphi, & T \geq T^{\text{ORX}}, \\ \varphi^{trunc}, & T < T^{\text{ORX}}. \end{cases}$$

See also below to see how the adapted LR looks like.

Another important difference concerns the a priori selection of the data. External consortium data combines many different banks and therefore different risk profiles than the one of our

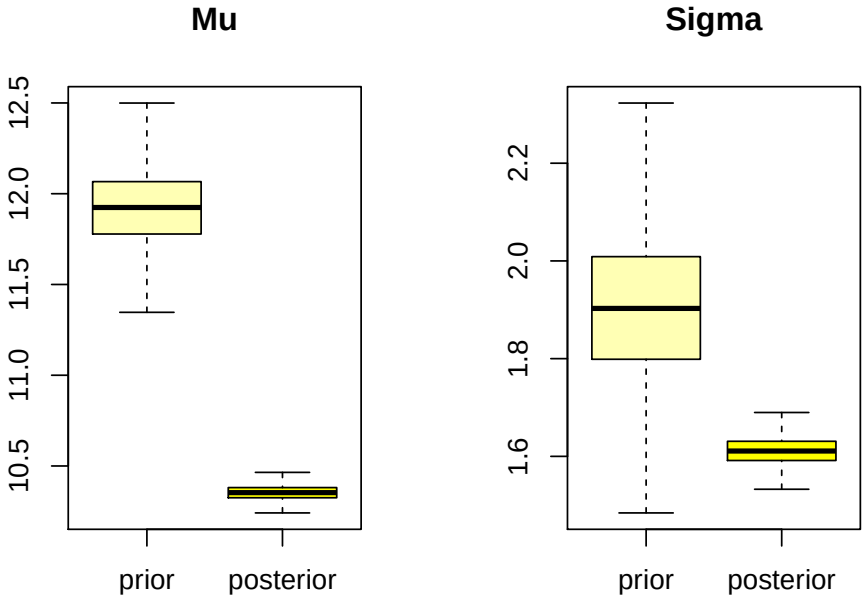

Figure 3.6: *After inclusion of external data, parameter uncertainty is reduced to a minimum.*

bank. In general losses are made anonymous, but they come with some labels like income size (*Small*, *Medium* or *Large*) and region (*Western Europe*, *Eastern Europe*, *North America* ...). With the help of our operational risk experts we determined a procedure to match ORX data with our risk profile.

Because our business in Asia or North America is marginal, we only consider data from the regions Western Europe and Eastern Europe . Furthermore we exclude all losses from Western Europe that do not match our income size. In Eastern Europe we decided to not exclude losses by income size because we are a major player in that region and the income size, which is calculated globally, would only distort this fact.

Another distortion factor we have to consider is that RBI has its major operational risk drivers in Eastern Europe (61 percent of the gross income and 85 percent of all operational losses), whereas in the ORX data base Eastern Europe is under represented (only 3 percent of all

losses). We tackle this problem by an adaption of the likelihood ratio LR:

$$\frac{\left(\prod_i \varphi^{ext}\left(\log(x_i^{west}-T)\,,\mu_t,\sigma_t\right)^{\lambda^{west}} \prod_j \varphi^{ext}\left(\log(x_j^{east}-T),\mu_t,\sigma_t\right)^{\lambda^{east}}\right)\cdot \pi^{ext}(\mu_t,\sigma_t)}{\left(\prod_i \varphi^{ext}\left(\log(x_i^{west}-T),\mu_{t-1},\sigma_{t-1}\right)^{\lambda^{west}} \prod_j \varphi^{ext}\left(\log(x_j^{east}-T),\mu_{t-1},\sigma_{t-1}\right)^{\lambda^{east}}\right)\cdot \pi^{ext}(\mu_{t-1},\sigma_{t-1})}, \tag{3.2}$$

i.e. we *reweight* the likelihood function to correct for the regional bias. The weights λ are chosen to match our risk profile:

$$\lambda^{west} = p_{west}\cdot\frac{n_{ORX}^{west}+n_{ORX}^{east}}{n_{ORX}^{west}},\quad \lambda^{east} = (1-p_{west})\cdot\frac{n_{ORX}^{west}+n_{ORX}^{east}}{n_{ORX}^{east}},\quad p_{west}=\frac{n_{int}^{east}}{n_{int}^{west}+n_{int}^{east}}$$

where n^{east}, n^{west} is the respective amount of data in the ORX or internal data.

With these adaptions, we again start our MCMC procedure.
To finally arrive at estimates for μ and σ in (3.1), we use the MMSE estimate, i.e. the mean of the MCMC samples.

Scenarios

> The problem with experts is that they do not know what they do not know.
>
> Nicholas Nassim Taleb, The Black Swan

Our business experts deliver scenarios of the following form:

Event Type	Scenario Short Name	Expected Annual Frequency	Expected Median Case	Reasonable Worst Case
EL06	*Failure of primary data centre (environmental factors)*	0,02	20.000,00	1.000.000,00

- **Expected Annual Frequency** How often the scenario is expected to happen in a year, denoted by λ_i
- **Median** The 50% - quantile of the scenario loss, denoted by Q^i_{50}
- **Reasonable Worst Case** The 99% - quantile of the scenario loss, denoted by Q^i_{99}

Because the threshold in our model is in most cases much higher than the median, we will not explicitly use the median estimate of the expert in our model. However, we need the information for benchmarking the risk of the particular scenario and to be able to give immediate feedback to the expert regarding the scenario.

Given σ and the 99% - quantile given by the i-th scenario we can calculate the distribution parameter μ in our body/tail model by:

$$\mu(Q^i_{99}, \sigma) = \log(Q^i_{99} - T) - \Phi^{-1}\left(1 - 0.01/\left(1 - \hat{F}(T)\right)\right) \cdot \sigma.$$

Therefore we can translate every scenario into an opinion of the expert on the distribution parameter μ. Furthermore, we decided that an high expected frequency of the scenario should be considered in the model by an increased weight of the resulting scenario estimate of μ.

To finally incorporate the scenario in our final capital number we make the following assumption: The distribution of μ is a mixture of the posterior distribution given by our internal/external data and the forward looking estimates of the experts. The weights are given by the proportion of the loss frequency these estimates represent. For σ however, we assume that the distribution is fully determined by our data. If we denote the MMSE estimators of the Two-Stage Bayes approach by $\mu^{\text{Data}}, \sigma^{\text{Data}}$, our final MMSE estimator including the scenario analyses are given by $\hat{\sigma} = \sigma^{\text{Data}}$ and

$$\hat{\mu} = \frac{1}{\lambda} \cdot \left(\lambda^{\text{Data}} \mu^{\text{Data}} + \sum_{i=1}^{n_{\text{scen}}} \lambda_i \mu(Q^i_{99}, \sigma^{\text{Data}}) \right) \text{ with } \lambda = \lambda^{\text{Data}} + \sum_{i=1}^{n_{\text{scen}}} \lambda_i.$$

Tail Threshold

Changing the threshold T has the following effect on the estimator $\hat{\theta}_T = (\mu, \sigma)$:

DECREASING $\searrow$: More data in the tail, decreasing variance of estimator, but data is less homogeneous, bias increases

INCREASING $\nearrow$: Less data in the tail, more homogeneous data and therefore smaller bias, but at the cost of a higher estimator variance

It is not a simple task to have a small estimator variance $\mathrm{Var}(\hat{\theta}_T)$ and estimator bias $\mathrm{Bias}(\hat{\theta}_T)$ at the same time. A possible idea to find an optimal threshold is to minimize the sum of both with respect to the threshold, i.e. minimizing the Mean Squared Error (MSE):

$$\hat{T} = \mathrm{argmin}_T \, \mathrm{MSE}(\hat{\theta}_T) = \mathrm{Var}(\hat{\theta}_T) + \left(\mathrm{Bias}(\hat{\theta}_T)\right)^2 = \mathrm{E}\left[(\hat{\theta}_T - \theta_T)^2\right],$$

where θ_T denotes the "true" value of the parameter.

This idea to estimate the distribution threshold has been brought up by many authors, see e.g. [DdHPdV01], [GO01], [FdHP03]. Also have a look at [SM12] for a good overview of some similar methods. These papers deal with the estimation of the threshold in the Pareto case, and obtain the MSE by bootstraping the parameter hyperdistribution. However, in a Bayesian framework like it is used in the Two-Stage Bayes approach, it is possible to just use the MCMC sample chain to estimate the MSE.

Estimation will be based on two points:

- For calculating the MSE, the sample chain *after the first step* of the MCMC procedure (after inclusion of internal data) is used
- As "true" parameter the MMSE parameter *after the second step* (inclusion of external) is used.

In this way both the bias of the internal data parameter with regard to the external data is minimized as well as the variance of the internal data parameter.

This means the MSE for each threshold is estimated by

$$\frac{1}{n}\sum_{i=1}^{n}\left(\mu_i(T) - \mu^{\mathrm{ext}}(T)\right)^2 + \frac{1}{n}\sum_{i=1}^{n}\left(\sigma_i(T) - \sigma^{\mathrm{ext}}(T)\right)^2.$$

However, blindly taking the threshold that minimizes the MSE can take to numerical problems (see Figure 3.7). A solution is to not only rely on the MSE criterion, but to use several other measures as well:

- **Stability of Parameters:** A small change of the threshold should not change the model parameters significantly. We measure the parameter stability in terms of 99.9% quantile of the compound loss distribution.

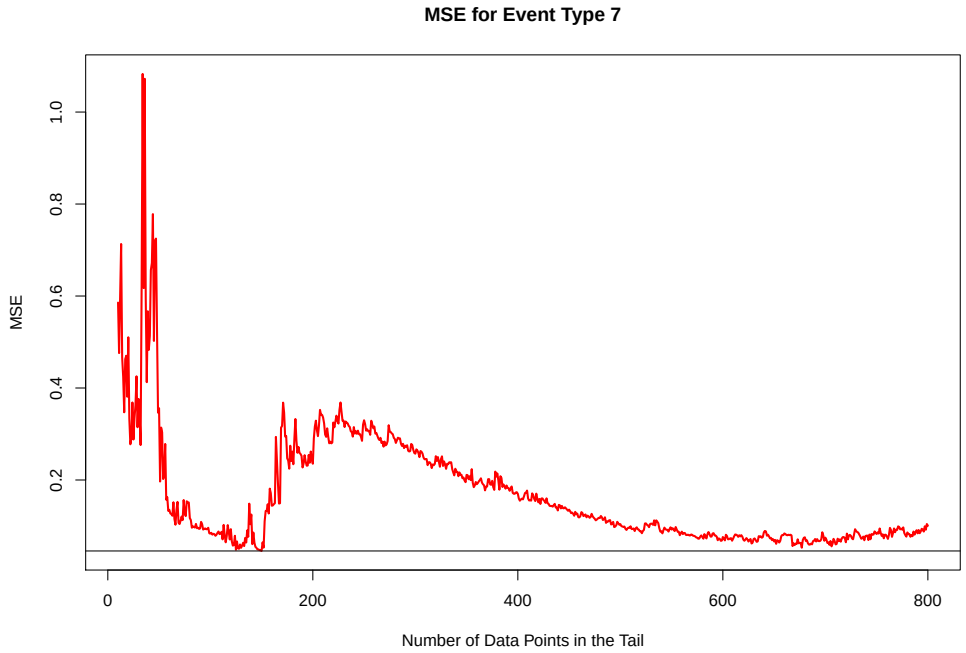

Figure 3.7: *The calculation of the MSE also highly depends on the underlying data. In the figure above we see that due to some data issues (two high severity losses have very similar height, which is numerically not optimal when using a shifted log normal) the MSE as a function of the threshold looks highly unstable. The global minimum is at about 150 data points in the tail - but would it not be more optimal to use the area around 700 data points, where the MSE seems to be more stable and comparably low?*

- **Goodness of Fit:** As we assume a log normal tail distribution, the data above T should have a reasonable good distribution fit. We measure the goodness of fit (GoF) with the Anderson-Darling test p-value. We plot both the quantile estimate and the GoF measure on a plot as a function of the number of data points in the tail.
- **Enough Data Points for Tail Distribution Estimation:** One important point is still to have enough data points in the tail for reasonable estimation results. For this reason, given that two different threshold are very similar in term of the above criteria, one should always choose the smaller one.

The final procedure to determine the tail threshold is based on the following (see also Figure 3.8):

(i) Based on the above criteria, ”stable” regions for the threshold are defined.

(ii) Within these regions, the threshold that minimizes the MSE is chosen.

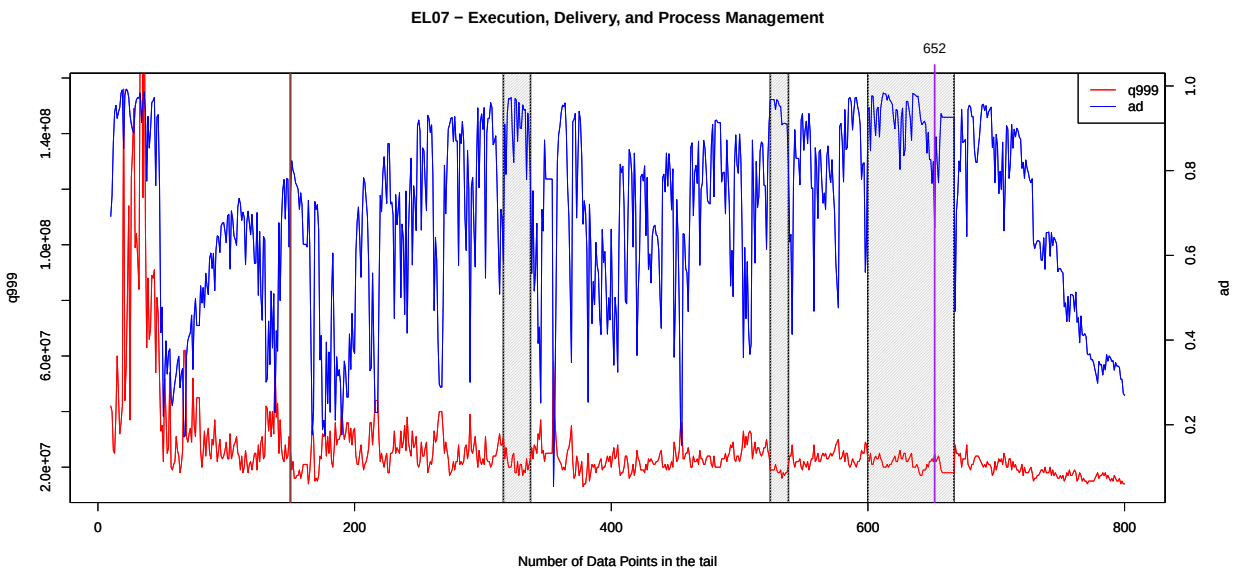

Figure 3.8: *Via a graphical assessment* stable regions *are defined, i.e. regions, where the goodness of fit (measured by the Anderson Darling p-value, see also Section 4.2) and the quantile estimates do not seem to fluctuate too much. Within these regions, the threshold that minimizes the MSE is determined (purple line, the brown line shows the global minimum of the MSE).*

Results

An advantage of the Two-Stage Bayes approach is that we can easily benchmark our internal data with external data: With the distribution of the parameters given internal data we can also derive a distribution of the OpVaR, as it only depends on the threshold and the two parameters. In this way we can compare the final OpVaR number after inclusion of external data to the range of possible numbers before inclusion and derive a confidence level.

	Stage 1	Stage 2	with Scen.
EL01 - Internal Fraud	[redacted]	[redacted]	[redacted]
EL02 - External Fraud	[redacted]	[redacted]	[redacted]
EL03 - Employment Practices and Workplace Safety	[redacted]	[redacted]	[redacted]
EL04 - Clients, Products and Business Practices	[redacted]	[redacted]	[redacted]
EL05 - Natural Disasters and Public Safety	[redacted]	[redacted]	[redacted]
EL06 - Technology and Infrastructure Failures	[redacted]	[redacted]	[redacted]
EL07 - Execution, Delivery, and Process Management	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]

Table 3.1: *OpVaR Numbers (EUR Mio) based on the MMSE estimators of the posterior distribution of μ and σ after each stage in the Bayesian approach. Numbers are censored to ensure data privacy.*

	μ	σ	Threshold T
EL01 - Internal Fraud	9.97	1.97	10000
EL02 - External Fraud	10.30	1.64	7000
EL03 - Employment Practices and Workplace Safety	8.96	1.66	4000
EL04 - Clients, Products and Business Practices	11.65	1.95	45000
EL05 - Natural Disasters and Public Safety	6.38	1.74	2000
EL06 - Technology and Infrastructure Failures	10.27	1.85	2000
EL07 - Execution, Delivery, and Process Management	12.13	1.71	2000

Table 3.2: Final Parameters for the Severity distribution

	rho.int	rho.ext
EL01 - Internal Fraud	0.15	0.015
EL02 - External Fraud	0.05	0.002
EL03 - Employment Practices and Workplace Safety	0.10	0.010
EL04 - Clients, Products and Business Practices	0.05	0.002
EL05 - Natural Disasters and Public Safety	0.01	0.006
EL06 - Technology and Infrastructure Failures	0.12	0.006
EL07 - Execution, Delivery, and Process Management	0.40	0.003

Table 3.3: *Parameters for* Σ

3.4 Dependency Modelling Step

To derive the final capital requirement for the bank, it is also important to have a look at the dependencies between the ORC. Assuming that the ORC are independent and every frequency distribution is Poisson, a simple formula makes it possible to calculate the OpVaR, because in this case the total loss sum

$$\sum_{i=1}^{7}\sum_{k=1}^{N_i} X_{i,k}$$

has the same distribution as

$$\sum_{k=1}^{\hat{N}} \hat{X}_k$$

where $\hat{N}$ is Poisson, where the parameter λ equals the sum of the Poisson rates λ_i of the individual ORC. The severity $\hat{X}$ is distributed with the mixture distribution of the severity distributions F_i, i.e.

$$\hat{X} \sim F(.) = \frac{1}{\lambda}\sum_{i=1}^{7} \lambda_i F_i(.).$$

This can be shown by using the formula for the characteristic function of the compound loss S and the convolution property of the characteristic function.

The other extreme of dependency is full dependence. A simple heuristic is that full dependence of the underlying frequency distributions, in a sense that $N_1 = N_2 = \cdots = N_7$, implies that the bank's OpVaR is "not very different" to the sum of the individual single-ORC OpVaR ([BK] showed

an asymptotic equality). The sum can be seen as a conservative estimate of the total OpVaR, but one has to be careful: in general the subadditivity inequality

$$\mathrm{OpVaR}_{\mathrm{total}} := \mathrm{OpVaR}\left(\sum_{i=1}^{7}\sum_{k=1}^{N_i} X_{i,k}\right) \leq \sum_{i=1}^{7} \mathrm{OpVaR}\left(\sum_{k=1}^{N_i} X_{i,k}\right) =: \mathrm{OpVaR}_{\mathrm{cons}}$$

does NOT hold, especially in cases of distributions with fat tails. See also [NECD06] for an analysis of this matter. However, amongst operational risk modellers dependency modelling is seen as illustrated in Figure 3.9: $\mathrm{OpVaR}_{\mathrm{cons}}$ as an upper bound of the capital requirement (and according to [oBS11] the obligatory capital requirement if the bank cannot show the soundness of its dependency modelling step) and the independent case as a lower bound.
Neither of the extreme dependency cases are realistic - the true total capital requirement will lie in between. For this particular model, the following decisions had been taken:

- Dependency is modelled between the compound losses $S_i = \sum_{k=1}^{N_i} X_{i,k}$.
- We use a t-copula to model the dependency. As described in Section 2.5, this makes it possible to include tail dependency. We use monthly data to estimate the t-copula.

3.4.1 Results

The final results for the OpVaR are calculated via a MC simulation.

Dependence Assumption	OpVaR	Addon
independence	[redacted]	0%
Gaussian copula	[redacted]	[redacted]
t-copula	[redacted]	[redacted]
total dependence	[redacted]	[redacted]

Table 3.4: *OpVaR Numbers (EUR Mio) using different assumptions for the dependence structure. The addon is calculated based on the ideal case of independence. Numbers are censored to ensure data privacy.*

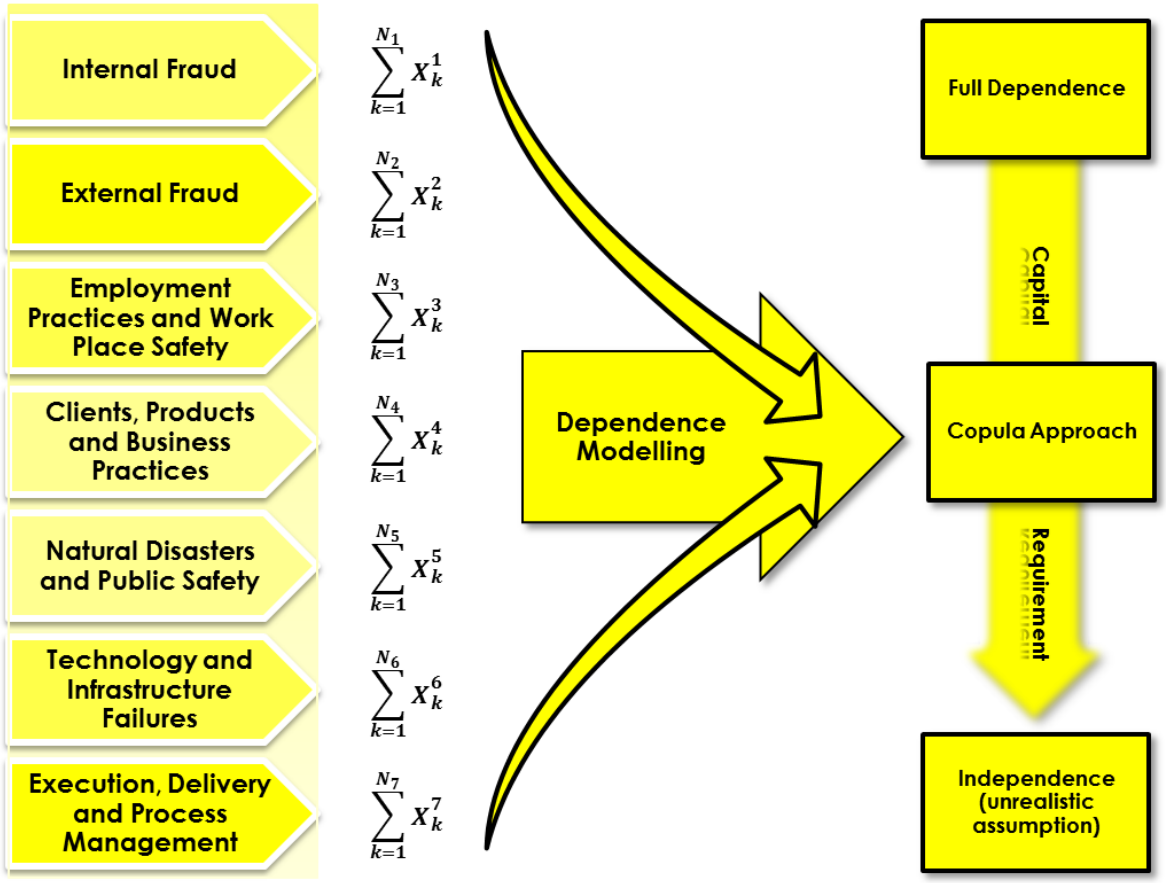

Figure 3.9: *The dependence structure has a significant impact on the final capital requirement.*

3.5 Allocation of Capital

One major problem with the LDA is that changes in the management of operational risk will need much time until they can be reflected in the capital requirement. Therefore the capital requirement as such is not a good incentive for a better ORM. However, a bank with many different units (e.g. subsidiaries across the world) can use another mechanism for incentivization: the allocation of the capital to each of these units.

Simple allocation mechanism like allocation in terms of percentage of gross income are easy to implement, but they give no motivation to think about how to prevent operational losses. Therefore this model allocates the capital based on a score with the following points:

Data Quality Is the unit cooperative in terms of providing well documented loss data?

Scenario Quality Are the scenarios well structured and are the chosen experts reliable?

Frequency of Losses Did the unit produce more or less losses than others?

Frequency of High Losses How many losses above a threshold (EUR 1 Million) happened in the unit?

Another important part of the scorecard are the **Key Risk Indicators**. KRIs give a good picture of potential risk in the units. For this reason numbers like the staff fluctuation, number and duration of system failures play an important part in the final allocation algorithm.

Based on this scorecard the units get more or less capital than they would based on a simple gross income allocation. This gives the unit an incentive to look at the above points and to improve their ORM.

3.6 Open Issues

Although the model in its current version should reflect the risk profile of the bank appropriately, there are still open issues for future work. The most important are

- **Insurance:** The Basel II accord allows to decrease the capital requirement if the bank can show that it is insured against operational risk losses. The model should therefore not only include an insurance module, but also be able to show for which event types an insurance would be most reasonable to decrease the risk.
- **Explaining the Model to the Business:** A good model should not only be used for the capital requirement. It should also help to improve the ORM and show where the riskiest areas are. This can only take place if the model produces an explainable output that is trusted by the business experts. It is an open question how this could look like.

Chapter 4

Validation of the Model

> 'That's right,' shouted Vroomfondel, 'we demand rigidly defined areas of doubt and uncertainty!'
>
> Douglas Adams, The Hitchhiker's Guide to the Galaxy (1979)

Before a model can be approved by the regulator, it is obligatory to set up a process of **internal validation** of the model. [oBS11] explicitly states that a validation activity should include evalution of

- Distributional assumptions
- Dependency assumptions
- The four key elements of the AMA
- Qualitative aspects (e.g. internal controls, use test, reporting, documentation of the model, role of senior management and organisational aspects)

Even though qualitative aspects should play an important role in the validation to ensure a proper risk management of bank, this thesis will concentrate on the first three quantitative aspects of an internal model validation.

4.1 Testing the Distribution Assumptions

4.1.1 EDF Statistics

Many statistical tests to assess the goodness of fit of a distribution have been developed on the idea of measuring some kind of distance between the empirical distribution function (EDF) of the observations $x_1, ..., x_n$, defined as

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{x_i \leq x},$$

and a given theoretical cdf $F(.)$. According to the Glivenko-Cantelli Theorem (see [Can33]) and other, stronger results like the Kolmogorov theorem ([Kol33]) it is well known that, for an increasing sample size, $\hat{F}_n$ converges to "true" cumulative distribution function. Based on this idea, these measures of distance (also known as EDF statistics) should give a good picture of the goodness of fit.

The most popular of these statistics is the Kolmogorov-Smirnov test (KS test) statistic (see [Smi48]). It is defined as

$$KS = \sqrt{n} \sup |\hat{F}_n(x) - F(x)|.$$

However, the KS test has been shown not to be very powerful in some cases (see [Ste74]). Moreover, for operational risk models, it is particular important to use tests that assign a higher weight to the tail proportion of the distribution. An alternative to the KS test are statistics based on the measure (see [AD52])

$$n \int_{-\infty}^{\infty} (\hat{F}_n(x) - F(x))^2 \Psi(x) dF(x), \tag{4.1}$$

where $\Psi(.)$ is a nonnegative weighting function chosen by the experimenter. There are many proposals in literature how to chose $\Psi(.)$. Cramér (see [Cra28]) and von Mises (see [VM28]) independently proposed to use $\Psi_{CM} \equiv 1$, which is now known as the Cramér-von Mises (CvM) criterion .

However, as explained, for the purposes of tail modelling it would be useful to put weight on the tail. One test that fulfils this criterion is the Anderson-Darling test (AD test) (see [AD54]), which is based on

$$\Psi_{AD}(x) = \frac{1}{F(x)\,(1 - F(x))}.$$

The $F(x)$ and the $(1 - F(x))$ term ensure a high weight on both the lower and the upper tail of the distribution. A useful modification for operational risk, where the capital mainly depends on the upper tail, was suggested in [CRF05]:

$$\Psi_{AD^2}(x) = \frac{1}{(1 - F(x))^2}$$

[CRF05] also suggested a modification to the KS test:

$$KS^* = \sqrt{n} \sup |\sqrt{\Psi(x)}(\hat{F}_n(x) - F(x))|, \tag{4.2}$$

where $\Psi(.)$ can be used to weight the different parts of a distribution similiar as in (4.1), using for example Ψ_{AD} and Ψ_{AD^2}.

The computation of the statistics based on (4.2) (*supremum* type statistics) and (4.1) (*quadratic* type statistics) does not seem trivial. However, there are some computation formulas (see [CRF05] for their derivation, even for truncated distributions). See Table 4.1 for the computation formulas for the most important EDF statistics.

To be able to do statistical inference, we have to derive the p-values under the null hypothesis that the data indeed is distributed with cdf $F(.)$. Since there are in general no closed formulas for the distribution of the test statistics, it is necessary to *bootstrap* the p-values, i.e. to simulate test statistics under the null hypothesis and derive their empirical distribution.

Supremum type statistics: $\sqrt{n}\sup|\Psi(x)(\hat{F}_n(x)-F(x))|$

Weight $\Psi(x)$	***Computing formula***
KS: 1	$\sqrt{n}\max\left\{\sup_j\left\{\frac{j}{n}-y_j\right\},\sup_j\left\{y_j-\frac{j-1}{n}\right\}\right\}$
$\frac{1}{F(x)(1-F(x))}$	$\sqrt{n}\max\left\{\sup_j\left\{\frac{\frac{j}{n}-y_j}{\sqrt{y_j(1-y_j)}}\right\},\sup_j\left\{\frac{y_j-\frac{j-1}{n}}{\sqrt{y_j(1-y_j)}}\right\}\right\}$

Quadratic type statistics: $n\int_{-\infty}^{\infty}(\hat{F}_n(x)-F(x))^2\Psi(x)dF(x)$

Weight $\Psi(x)$	***Computing formula***
CvM: 1	$\frac{n}{3}+\frac{1}{n}\sum_{j=1}^{n}(1-2j)y_j+\sum_{j=1}^{n}y_j^2$
AD: $\frac{1}{F(x)(1-F(x))}$	$-n+\frac{1}{n}\sum_{j=1}^{n}(1-2j)\log y_j-\frac{1}{n}\sum_{j=1}^{n}(1+2(n-j))\log(1-y_j)$
$\frac{1}{(1-F(x))^2}$	$\frac{1}{n}\sum_{j=1}^{n}(1-2(n-j))\frac{1}{1-y_j}+2\sum_{j=1}^{n}\log(1-y_j)$

Table 4.1: *Computing formulas for EDF statistics (see [CRF05]). We denote the ordered observations by* $x_{(j)}$*, i.e.* $x_{(1)}\leq\cdots\leq x_{(n)}$*. Moreover we set* $y_j:=\hat{F}_n(x_{(j)})$*.*

4.1.2 Graphical Tools

Graphical tools are also very useful in the assessment of the goodness of fit of a distribution. In the case of a continuous distribution it can be very helpful to overlay the histogram of the observations with the theoretical density function.

The most popular graphical tool is the QQ-plot, which compares the empirical and the theoretical quantiles (see Figure 4.1). If the data is distributed according to the theoretical distribution, the points on QQ-plot are expected to form a line.

4.1.3 Distribution Assumptions

Distribution assumptions are made at several parts of the model:

- Distribution of the frequency (Poisson / negative binomial)
- Distribution of the severity tail (shifted log normal)
- Hyperdistribution of the severity distribution parameters μ and σ in the MCMC procedure (μ is normal and $1/\sigma^2$ is Gamma)

The frequency data is very scarce, therefore the validation of the frequency distribution assumption relies on the graphical assessment of the goodness of fit.

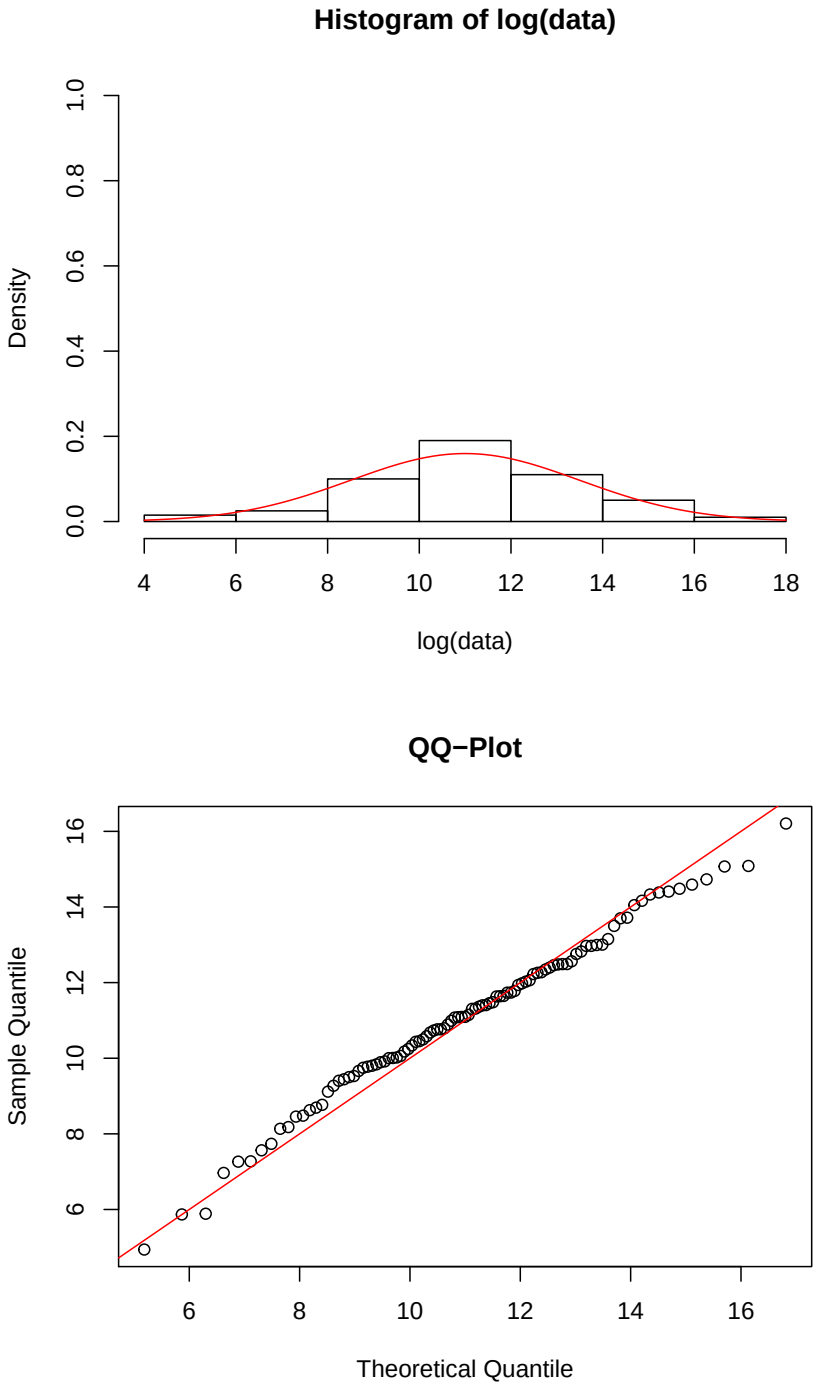

Figure 4.1: *Graphical tools to assess the goodness of fit. For the above plots a log normal distribution with parameters* $\mu = 11$ *and* $\sigma = 2.5$ *is assumed.*

To assess the goodness of fit of the severity tail distribution, we also have to bear in mind that the final parameter estimate depends on three different data elements (internal and external data, scenarios).

Scenarios are forward looking, therefore we should not include them into a backward looking statistical test. For that reason we define the null hypothesis as

$$\mathcal{H}_0 : X_i|_{X_i>T} \sim F(x) = \Phi\left(\frac{\log(x-T) - \mu^{\text{Data}}}{\sigma^{\text{Data}}}\right),$$

where μ^{Data} and σ^{Data} are the MSE estimates after inclusion of external data.
To come to conclusion whether the null hypothesis should be rejected, it is needed that both the graphical tools and the p-values of the mentioned statistics are considered and assessed. In particular, the AD test *should not reject with* $\alpha = 0.05$.

A similar validation of the hyperparameter distribution of the MCMC procedure after the first step (inclusion of internal data) is also undertaken. In this case, we define the null as

$$\mathcal{H}_0 : \mu \sim \Phi\left(\mu^{\text{hyper}}, \sigma^{\text{hyper}}\right), \quad \frac{1}{\sigma^2} \sim \text{Gamma}(\alpha^{\text{hyper}}, \beta^{\text{hyper}}),$$

where μ^{hyper}, σ^{hyper}, α^{hyper} and β^{hyper} are the parameters fitted to the realizations of the MCMC of the first internal data step.

4.2 Testing the Dependency Assumptions

4.2.1 Assessing the Dependence Structure

For the purpose of capital estimation, the most conservative assumption is perfect dependence (which implies that the OpVaR of the bank equals the sum of the individual ORC-OpVaR). Therefore it has to be validated whether the dependency is less than perfect.

For this reason graphical tools which are able to capture several dimensions are very useful. One idea could be to make a scatterplot (i.e. plotting data pairs on two axes of a plot) to analyze the dependence. However, it is recommended to use a scale-free graphical representation of the dependence (see [GF07]). This can be achieved by plotting the pair of ranks. Perfect dependence implies that the copula equals the Frechet upper bound (see [Fré51]), i.e. in two dimensions

$$C_{\text{Frechet}}(u_1, u_2) = \min\{u_1, u_2\}.$$

Figure 4.2 shows how simulations from a comontone copula looks like: the rank plots seems to form a line. The more dispersed the plot looks like, the less dependence can be assumed. Figure 4.2 also gives a picture of other forms of dependence that can be assessed by the pair of ranks.

4.2.2 Goodness of Fit of the Copula

After assessing the dependence in general, it would be of great interest to assess the goodness of fit of a particular copula. In the case of distributions, the histogram/density-plot and the QQ-plot was already mentioned. In the case of copulas, it is possible to produce similar pictures (see Figure 4.3):

- In the case of a continuous copula (like the Gaussian or t-copula), we also can plot a two-dimensional density in form of a contour plot.
- We can compare the values of the theoretical copula with the values of the empirical copula for a paired data set $(x_i^1, x_i^2), i = 1 \dots n$, which is defined as

  $$\hat{C}_n(u_1, u_2) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{x_i^1 \leq u_1, x_i^2 \leq u_2\}}.$$

  The result is a two dimensional "PP"-plot.

Normally one has to work with more than two dimensions. However, in this case the assessment can be made for all possible combination of pairs.

4.2.3 Dependency Assumptions

The dependency assumptions of the model that have to be tested are

- t-copula dependence structure between ORC

- Independence of the hyper parameter distributions of μ and σ after the first step in the MCMC procedure

For the latter point the rank plot and Kendall's tau will be assessed to see if independence is a reasonable assumption.

To validate the t-copula dependence structure between the ORC, the graphical tools are used to review the goodness of fit for the copula.

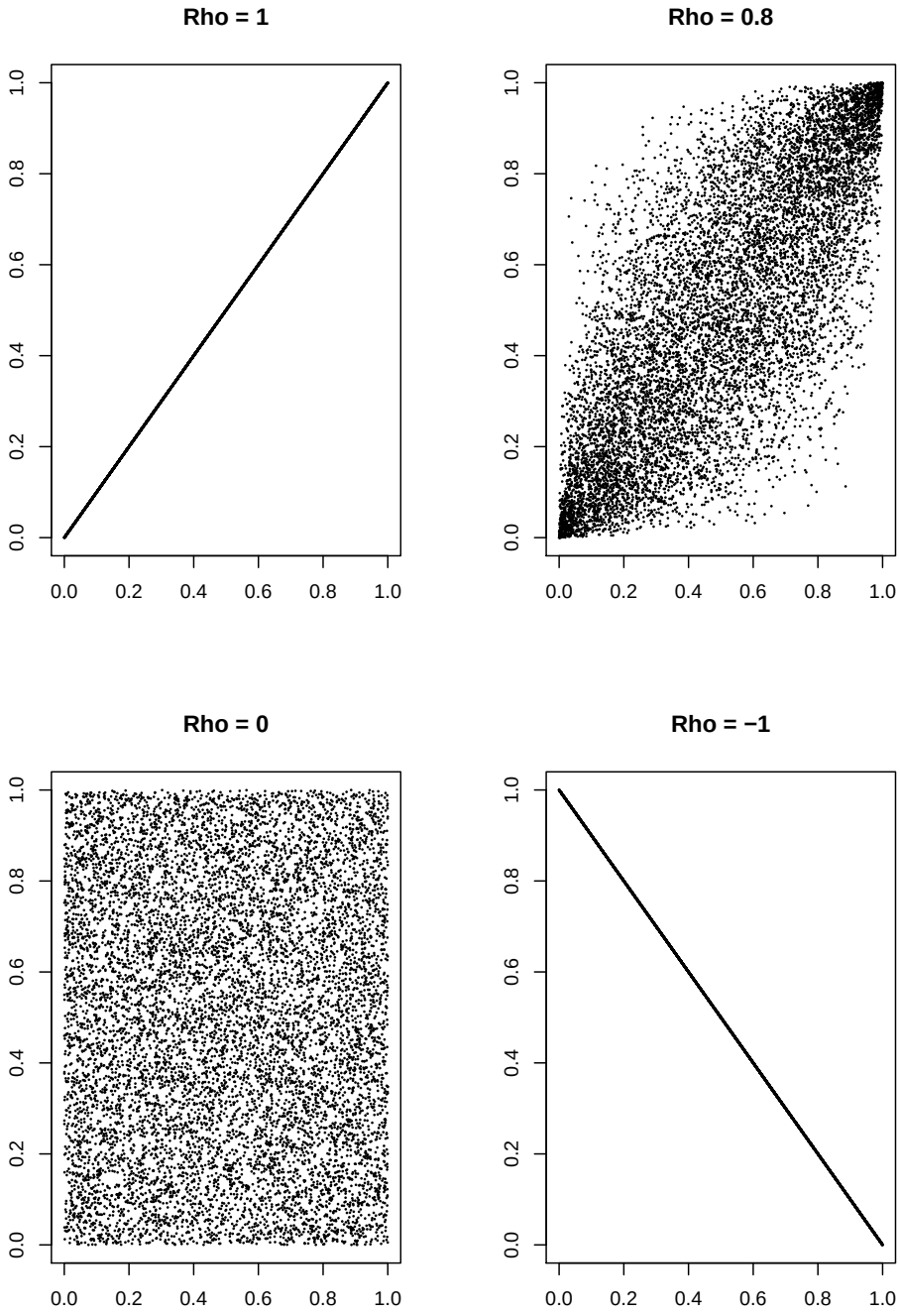

Figure 4.2: *The Gaussian copula gives a good picture of how perfect dependence / independence / perfect negative dependence looks like. The correlation parameter $\rho \in [-1, 1]$ interpolates between perfect dependence ($\rho = 1$), independence ($\rho = 0$) and perfect negative dependence ($\rho = -1$). The above plots show 10,000 simulations from a Gaussian copula with different values for ρ.*

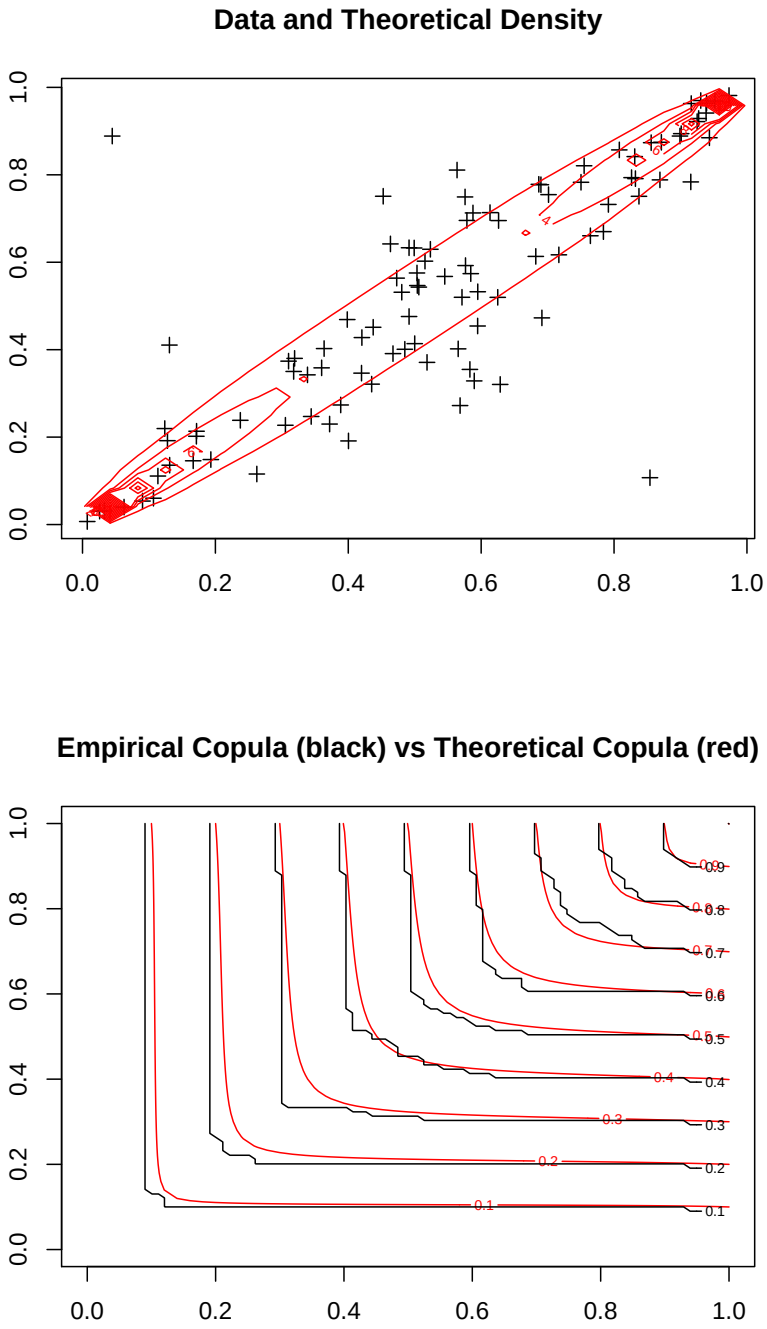

Figure 4.3: *Graphical tools to assess the goodness of fit of the copula. For the above plots a t-Copula with* $df = 1$ *and a positive correlation of 0.9 is assumed.*

4.3 Reviewing the Key Elements of the AMA

The last requirement for the quantitative model validation is to review the use of the four key elements (internal data, external data, scenarios, BEICF).

The inclusion of the key elements is a requirement of the AMA approach. However, there is no prescribed way how to include them. The Bayesian framework offers a possible solution to this problem: internal data is treated as the starting point for the estimation, external data is used to lower the uncertainty of the parameter. Scenarios (that also incorporate BEICF in their estimation) are then used to adapt the estimate and to include a forward looking aspect into the model.

For validation purposes, the questions are:

- Does every source have a significant impact on the OpVaR?
- If yes, how strong is the influence of each element?

The Bayesian framework of the Two-Stage Bayes approach can also be of great use for that matter. With an analysis of the parameter hyper distributions after each step of data inclusion (see e.g. Figure 3.6 or Figure 3.2), the impact of each key element becomes visible. A future challenge still remains: What is the optimal impact of each data element? Should this be chosen by a model, or even an expert? In the end it remains difficult to pinpoint the operational risk profile of an individual credit institute, it is a task that has to be done with the help of the individual stake holders in the bank.

Chapter 5

Conclusion

> ...and go on till you come to the end: then stop.
>
> Lewis Carroll, Alice in Wonderland

The development of the AMA capital model bears many challenges. The described approach was developed to met some of them:

- **Many different inhomogeneous risk drivers:** The model uses a multivariate approach with seven different ORC. A copula is used to capture the dependence structure within the different model cells.
- **Fat tails of the loss distributions:** Although it is very hard or even impossible to capture tail risk with a model, many measures were taken to reduce the possibility of missing the identification of a high risk area. First of all the body-tail approach ensures a high concentration of the modelling effort on the tail. Moreover, inclusion of the key elements via the Two-Stage Bayes approach provides another sources of data to point out tail risk.
- **Robust, yet risk sensitive OpVaR:** A good model should react to changes in the operational risk, nevertheless it should be robust to insignificant data changes. The Bayesian MCMC procedure can be seen as a robust way to estimate the severity distribution (see [ZGFT13]). Moreover there is the approach for the threshold choice that guarantees for a stable fit of the severity distribution and therefore ensures stability of the model.
- **Validation of the model:** As shown in the previous chapter, the model is not just used as a black box that outputs some capital number. Each important model assumption is thoroughly validated and tested via a validation framework.

Yet, it has to be clear that a model can produce a capital number, but managing operational risk is so much more than just looking at one number.
It is about defining sufficient control mechanisms to prevent future losses. It is about raising the awareness for hidden risks in the daily business.

Sound operational risk management requires the cooperation of every employee of the institute. Everyone can help to prevent fraud by reporting suspicious behaviour. Everyone has to be honest to their clients to prevent legal cases due to flawed advice. Everyone can ensure that they are fit for their work and help to decrease the number of executional errors.

A model can be a good start to discuss the operational risk in a bank.

But it is definitely not the end.

Chapter 6

Appendix

6.1 Codes

> I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs.
>
> Maurice Wilkes (1949)

The following functions were were written in the programming language `R`. Besides the base packages, the following packages are needed to run the codes:

- `actuar` ([DGP08]): provides a very efficient distribution discretizing algorithm.
- `laeken` ([AT13]): robust estimates to fit the pareto distribution.
- `maxLik` ([HT11]): very efficient algorithm for maximum likelihood optimization
- `evir`: provides the cdf of the generalized Pareto distribution
- `mvtnorm` ([GB09]): provides the multivariate Normal and t-distribution

For every described function, a full documentation (in terms of the usual `R`-help file documentation and the code) is given. The functions are categorized in the two most challenging tasks: distribution fitting for the severity and numerical calculation of the OpVaR.

6.1.1 Fitting of the Severity Distribution

`MCMC` *Function to fit a truncated distribution via Monte Carlo Markov Chain Methods.*

Description

MCMC fits a truncated log normal distribution and returns samples from the posterior distribution of the parameters

Usage

```
MCMC(data, threshold,
  Sigma = 0.05 * matrix(c(1, 0, 0, 0.5), ncol = 2),
  n = 1e+05, weights = NULL, prior.method = "Jeffrey",
  mean.par = NULL, sigma.par = NULL, shape.par = NULL,
  rate.par = NULL, mu.int = NULL, sigma.int = NULL,
  start = c(12, 2), burnin = 0.5)
```

Arguments

`data`	Positive loss data
`threshold`	Threshold for the truncated distribution
`Sigma`	Covariance matrix for the MCMC random walk
`n`	number of iterations in the MCMC procedure
`weights`	If not NULL, weighted maximum likelihood will be performed.
`prior.method`	Either ”Jeffrey” (uninformed prior) or ”NormalGamma” (informed prior)
`mean.par`	Mean parameter of mu in ”NormalGamma”
`sigma.par`	Standard deviation parameter of mu in ”NormalGamma”
`shape.par`	Shape parameter of sigma in ”NormalGamma”
`rate.par`	Rate parameter of sigma in ”NormalGamma”
`mu.int`	Confidence interval of mu for restricted MCMC
`sigma.int`	Confidence interval of sigma for restricted MCMC
`start`	Start parameters for random walk in MCMC method
`burnin`	Percentage of sample that will be thrown away from the start

Value

Parameters, sample

Examples

```
body.data <- rlnorm(100,sdlog = 1.5, meanlog = 11)
tail.data <- rlnorm(100,sdlog = 2.5, meanlog = 11)
loss.data <- c(body.data[body.data <= 1e05], tail.data[tail.data > 1e05])
## Not run: mod <- MCMCfitter(loss.data, threshold = 0.8e05)
```

Code

```
function (data, threshold, Sigma = 0.05 * matrix(c(1, 0, 0, 0.5),
    ncol = 2), n = 1e+05, weights = NULL, prior.method = "Jeffrey",
    mean.par = NULL, sigma.par = NULL, shape.par = NULL, rate.par = NULL,
    mu.int = NULL, sigma.int = NULL, start = c(12, 2), burnin = 0.5)
{
    narrow <- !(is.null(mu.int) | is.null(sigma.int))
    if (is.null(weights))
        weights = rep(1, length(data))
    if (!prior.method %in% c("Jeffrey", "NormalGamma"))
        stop("Prior not defined!")
    if (prior.method == "Jeffrey") {
        prior <- function(mu, sigma, threshold) {
            if (threshold > -Inf) {
                u.star <- (threshold - mu)/sigma
                alpha <- dnorm(u.star)/(1 - pnorm(u.star))
                A = (1 - alpha^2 + u.star * alpha)/sigma^2
                B = alpha * (1 - u.star * alpha + u.star^2)/sigma^2
                C = (2 + u.star * alpha - alpha^2 * u.star^2 +
                  u.star^3 * alpha)/sigma^2
            }
            else {
                A = 1/sigma^2
                B = 0
                C = 2/sigma^2
            }
            return(sqrt(A * C - B^2))
        }
    }
    if (prior.method == "NormalGamma") {
        prior <- function(mu, sigma, threshold) {
            prior.sigma <- dinvgamma(sigma^2, shape = shape.par,
                scale = rate.par) * 2 * sigma
            prior.beta <- dnorm(mu, mean = mean.par, sd = sigma.par)
            return(prior.sigma * prior.beta)
        }
    }
    exp.data = data
    threshold = log(threshold)
```

```
data = log(exp.data[exp.data > exp(threshold)])
weights <- weights[exp.data > exp(threshold)]
rw <- rmvnorm(n, mean = c(0, 0), sigma = Sigma)
mu = start[1]
sigma = start[2]
acc = NULL
former.lhood = dnorm(data, mean = mu, sd = sigma)/(1 - pnorm(threshold,
    mean = mu, sd = sigma))
mchain = matrix(NA, ncol = 2, nrow = n)
colnames(mchain) = c("mu", "sigma")
for (i in 1:n) {
    cand.mu = mu + rw[i, 1]
    cand.sigma = sigma + rw[i, 2]
    if (cand.sigma <= 0) {
        mchain[i, 1] = mu
        mchain[i, 2] = sigma
        next
    }
    if (narrow) {
        if (cand.sigma > sigma.int[2] | cand.sigma < sigma.int[1] |
            cand.mu > mu.int[2] | cand.mu < mu.int[1]) {
            mchain[i, 1] = mu
            mchain[i, 2] = sigma
            next
        }
    }
    Tt = 1
    cand.lhood = dnorm(data, mean = cand.mu, sd = cand.sigma)/(1 -
        pnorm(threshold, mean = cand.mu, sd = cand.sigma))
    lhood.ratio = exp(sum(weights * (log(cand.lhood) - log(former.lhood))) +
        log(prior(cand.mu, cand.sigma, threshold)) - log(prior(mu,
        sigma, threshold)))
    alpha = min((lhood.ratio)^(1/Tt), 1)
    if (!is.nan(alpha)) {
        if (runif(1) < alpha) {
            mu = cand.mu
            sigma = cand.sigma
            former.lhood = cand.lhood
            acc = c(acc, TRUE)
        }
    }
    mchain[i, 1] = mu
    mchain[i, 2] = sigma
}
acc.rate = sum(acc)/n
```

```
    ma <- function(x, n.m = n/100) {
        filter(x, rep(1/n.m, n.m), sides = 2)
    }
    par(mfrow = c(2, 1), oma = c(0, 0, 2.5, 0))
    plot(1:nrow(mchain), mchain[, 1], "l", col = "grey", main = "Mu",
        xlab = "Samples", ylab = "")
    lines(1:nrow(mchain), ma(mchain[, 1]), col = "black")
    plot(1:nrow(mchain), mchain[, 2], "l", col = "grey", main = "Sigma",
        xlab = "Samples", ylab = "")
    lines(1:nrow(mchain), ma(mchain[, 2]), col = "black")
    burnin = round(burnin * n)
    mchain <- mchain[(burnin + 1):nrow(mchain), ]
    means <- apply(mchain, 2, mean)
    Sigma <- var(mchain)
    output <- list(acc.rate = acc.rate, means = means, Sigma = Sigma,
        sample = mchain)
    return(output)
}
```

`fittruncdistr` *Function to fit a truncated distribution to a given data sample.*

Description

fitttruncdistr does truncated maximum likelihood fitting for a given threshold function

Usage

```
fittruncdistr(data, threshold, distr, start = NULL,
  fix.arg = NULL, pos = TRUE, upper.threshold = Inf,
  method.pareto = "ml", weights = NULL)
```

Arguments

`data`	Positive loss data
`threshold`	Threshold for the truncated distribution
`distr`	string of a function with pdistr, qdistr and ddistr function, e.g. ”norm”
`start`	starting value for optimization. For the most common distributions it is not necessary to give a starting value
`fix.arg`	arguments which should be held fix in the optimization
`pos`	until now two different forms of optimization are implemented: arbitrary number of parameters, all positive (pos = TRUE) or two parameters, one arbitrary, they other one positive e.g. (pos = c(FALSE,TRUE))

`upper.threshold`
: fitting a distribution between two points on the real line

`method.pareto`
: method for finding the shape of pareto (distr = "pareto1"). Provided by package laeken. Can be one of c("ml", "Hill", "ISE", "QQ", "PDC", "WML").

`weights`
: If not NULL, weighted maximum likelihood will be performed.

Value

Parameters of the fit

Examples

```
body.data <- rlnorm(100,sdlog = 1.5, meanlog = 11)
tail.data <- rlnorm(100,sdlog = 2.5, meanlog = 11)
loss.data <- c(body.data[body.data <= 1e05], tail.data[tail.data > 1e05])
fittruncdistr(loss.data,distr = "lnorm", threshold = 1e05)
```

Code

```
function (data, threshold, distr, start = NULL, fix.arg = NULL,
    pos = TRUE, upper.threshold = Inf, method.pareto = "ml",
    weights = NULL)
{
    if (is.null(weights))
        weights <- rep(1, length(data))
    if (length(weights) != length(data))
        stop("Weights do not have the right length!")
    if (is.infinite(upper.threshold)) {
        weights <- weights[data > threshold]
        data <- data[data > threshold]
    }
    else {
        weights <- weights[data > threshold & data <= upper.threshold]
        data <- data[data > threshold & data <= upper.threshold]
    }
    pdistname <- paste("p", distr, sep = "")
    if (!exists(pdistname, mode = "function"))
        stop(paste("The ", pdistname, " function must be defined"))
    ddistname <- paste("d", distr, sep = "")
    if (!exists(ddistname, mode = "function"))
        stop(paste("The ", ddistname, " function must be defined"))
    if (is.null(start)) {
        if (distr == "norm") {
            n <- length(data)
            sd0 <- sqrt((n - 1)/n) * sd(data)
            mx <- mean(data)
```

```
    start <- list(mean = mx, sd = sd0)
}
if (distr == "lnorm") {
    if (any(data <= 0))
        stop("values must be positive to fit a lognormal distribution")
    n <- length(data)
    ldata <- log(data)
    sd0 <- sqrt((n - 1)/n) * sd(ldata)
    ml <- mean(ldata)
    start <- list(meanlog = ml, sdlog = sd0)
    pos = c(FALSE, TRUE)
}
if (distr == "llogis") {
    ldata <- log(data)
    n <- length(ldata)
    m <- thetaHill(data, x0 = threshold)
    v <- (n - 1)/n * var(ldata)
    start <- list(shape = m, rate = sqrt(3 * v)/pi)
}
if (distr == "burr") {
    m <- mean(data)
    m2 <- mean(data^2)
    theta1 = 1 + sqrt(m2/(m2 - m^2))
    theta2 = m2/m * (1 - sqrt((m2 - m^2)/m2))
    start <- list(shape1 = theta1, shape2 = theta1, scale = theta2)
}
if (distr == "pareto1") {
    if (method.pareto == "Hill") {
        return(list(estimate = c(shape = thetaHill(data,
          x0 = threshold), min = threshold), maximum = NA))
    }
    if (method.pareto == "ISE") {
        return(list(estimate = c(shape = thetaISE(data,
          x0 = threshold), min = threshold), maximum = NA))
    }
    if (method.pareto == "PDC") {
        return(list(estimate = c(shape = thetaPDC(data,
          x0 = threshold), min = threshold), maximum = NA))
    }
    if (method.pareto == "QQ") {
        return(list(estimate = c(shape = thetaQQ(data,
          x0 = threshold), min = threshold), maximum = NA))
    }
    if (method.pareto == "WML") {
        return(list(estimate = c(shape = thetaWML(data,
```

```
          x0 = threshold), min = threshold), maximum = NA))
    }
    m <- mean(data)
    m2 <- mean(data^2)
    theta1 = 1 + sqrt(m2/(m2 - m^2))
    start <- list(shape = theta1, min = threshold)
}
if (distr == "pareto" | distr == "pareto2") {
    m <- mean(data)
    m2 <- mean(data^2)
    theta1 = 1 + sqrt(m2/(m2 - m^2))
    theta2 = m2/m * (1 - sqrt((m2 - m^2)/m2))
    start <- list(shape = theta1, scale = theta2)
}
if (distr == "gpd") {
    gpd.mod <- gpd(data, threshold = threshold)
    return(list(estimate = (c(unlist(gpd.mod$par.ests),
        mu = threshold)), maximum = -gpd.mod$nllh.final))
}
if (distr == "pois") {
    start <- list(lambda = mean(data))
}
if (distr == "exp") {
    start <- list(rate = 1/mean(data) * 0.001)
}
if (distr == "gev") {
    n <- length(data)
    sigma0 <- sqrt(6 * var(data))/pi
    mu0 <- mean(data) - 0.57722 * sigma0
    xi0 <- 0.1
    theta <- c(xi0, sigma0, mu0)
    negloglik <- function(theta, tmp) {
        y <- 1 + (theta[1] * (tmp - theta[3]))/theta[2]
        if ((theta[2] < 0) || (min(y) < 0))
          out <- 1e+06
        else {
          term1 <- length(tmp) * logb(theta[2])
          term2 <- sum((1 + 1/theta[1]) * logb(y))
          term3 <- sum(y^(-1/theta[1]))
          out <- term1 + term2 + term3
        }
        out
    }
    fit <- optim(theta, negloglik, hessian = TRUE, tmp = data)
    if (fit$convergence)
```

```
        warning("optimization may not have succeeded")
    par.ests = list(xi = fit$par[1], mu = fit$par[3],
        sigma = fit$par[2])
    return(list(estimate = (c(unlist(par.ests))), maximum = -fit$value))
}
if (distr == "gamma") {
    n <- length(data)
    m <- mean(data)
    v <- (n - 1)/n * var(data)
    start <- list(shape = m^2/v, rate = m/v)
}
if (distr == "nbinom") {
    n <- length(data)
    m <- mean(data)
    v <- (n - 1)/n * var(data)
    size <- if (v > m)
        m^2/(v - m)
    else 100
    start <- list(size = size, mu = m)
}
if (distr == "geom") {
    m <- mean(data)
    prob <- if (m > 0)
        1/(1 + m)
    else 1
    start <- list(prob = prob)
}
if (distr == "beta") {
    if (any(data < 0) | any(data > 1))
        stop("values must be in [0-1] to fit a beta distribution")
    n <- length(data)
    m <- mean(data)
    v <- (n - 1)/n * var(data)
    aux <- m * (1 - m)/v - 1
    start <- list(shape1 = m * aux, shape2 = (1 - m) *
        aux)
}
if (distr == "weibull") {
    m <- mean(log(data))
    v <- var(log(data))
    shape <- 1.2/sqrt(v)
    scale <- exp(m + 0.572/shape)
    start <- list(shape = shape, scale = scale)
}
if (distr == "logis") {
```

```
        n <- length(data)
        m <- mean(data)
        v <- (n - 1)/n * var(data)
        start <- list(location = m, scale = sqrt(3 * v)/pi)
    }
    if (distr == "cauchy") {
        start <- list(location = median(data), scale = IQR(data)/2)
    }
    if (distr == "unif") {
        start <- list(min = 0, max = 1)
    }
}
if (is.infinite(upper.threshold)) {
    fnobj <- function(par) {
        z.H = 1 - do.call(pdistname, c(list(threshold), as.list(par),
            as.list(fix.arg)))
        NlogL = sum(weights * log(do.call(ddistname, c(list(data),
            as.list(par), as.list(fix.arg)))/z.H))
        if (is.na(NlogL))
            NlogL = -.Machine$double.xmax
        return(NlogL)
    }
}
else {
    fnobj <- function(par) {
        z.H = do.call(pdistname, c(list(upper.threshold),
            as.list(par), as.list(fix.arg))) - do.call(pdistname,
            c(list(threshold), as.list(par), as.list(fix.arg)))
        NlogL = sum(weights * log(do.call(ddistname, c(list(data),
            as.list(par), as.list(fix.arg)))/z.H))
        if (is.na(NlogL))
            NlogL = -.Machine$double.xmax
        return(NlogL)
    }
}
n.par <- length(start)
if (length(pos) == 1) {
    if (pos == TRUE) {
        A <- diag(n.par)
        B <- rep(0, n.par)
    }
}
else {
    if (length(pos) == 2) {
        A <- matrix(as.numeric(pos), 1, 2)
```

```
            B <- 0
        }
        else {
            A <- diag(pos)[pos, ]
            B <- numeric(sum(pos))
            if (sum(pos) == 1)
                A <- t(as.matrix(A))
        }
    }
    opttryerror <- try(mod <- maxLik(logLik = fnobj, start = unlist(start),
        constraints = list(ineqA = A, ineqB = B), method = "BFGS"),
        silent = TRUE)
    if (inherits(opttryerror, "try-error")) {
        mod <- optim(unlist(start), function(x) {
            -fnobj(x)
        })
        mod$estimate <- mod$par
        mod$maximum <- -mod$value
    }
    return(mod)
}
```

`gof_tail_for_fitter` *Function to asssess the fit of a truncated distribution.*

Description

Returns statistics and several plots to assess the fit of a truncated distribution.

Usage

```
gof_tail_for_fitter(data, threshold, distr, pars,
  log.scale = FALSE, upper.threshold = Inf)
```

Arguments

`data`	Positive loss data
`threshold`	Threshold where the tail starts
`distr`	String of a function with pdistr, qdistr and ddistr function, e.g. ”norm”
`pars`	Parameters of distr can be added (as a list)
`log.scale`	Indicates whether the distribution should be tested on the logarithm of the data
`upper.threshold`	Possible upper threshold of the distribution

Value

Different distribution tests, a QQ Plot and a histogram

Examples

```
body.data <- rlnorm(100,sdlog = 1.5, meanlog = 11)
tail.data <- rlnorm(100,sdlog = 2.5, meanlog = 11)
loss.data <- c(body.data[body.data <= 1e05], tail.data[tail.data > 1e05])
gof_tail_for_fitter(loss.data, threshold = 1e05,
distr = "lnorm", pars = list(meanlog = 11, sdlog = 2.5))
```

Code

```
function (data, threshold, distr, pars, log.scale = FALSE, upper.threshold = Inf)
{
    if (log.scale) {
        data <- log(data[data > threshold])
        threshold <- log(threshold)
    }
    else {
        data <- data[data > threshold]
    }
    data <- data[order(data)]
    pdistname <- paste("p", distr, sep = "")
    if (!exists(pdistname, mode = "function"))
        stop(paste("The ", pdistname, " function must be defined"))
    ddistname <- paste("d", distr, sep = "")
    if (!exists(ddistname, mode = "function"))
        stop(paste("The ", ddistname, " function must be defined"))
    qdistname <- paste("q", distr, sep = "")
    if (!exists(qdistname, mode = "function"))
        stop(paste("The ", qdistname, " function must be defined"))
    tailcdf1 <- function(x) {
        as.numeric((do.call(pdistname, c(list(x), as.list(pars)))))
    }
    tailcdf <- function(x) {
        eval <- tailcdf1(x)
        if (!is.na(eval)) {
            return(eval)
        }
        else return(0)
    }
    tailcdf <- Vectorize(tailcdf)
    z.H <- tailcdf(threshold)
    if (is.infinite(upper.threshold)) {
        z.U <- 1
```

```
    taild1 <- function(x) {
        as.numeric((do.call(ddistname, c(list(x), as.list(pars)))))
    }
    taild <- function(x) {
        eval <- taild1(x)/(1 - z.H)
        if (!is.na(eval)) {
            return(eval)
        }
        else return(0)
    }
    taild <- Vectorize(taild)
    tailq1 <- function(x) {
        as.numeric((do.call(qdistname, c(list(x), as.list(pars)))))
    }
    tailq <- function(q) {
        tailq1(q * (1 - z.H) + z.H)
    }
    tailq <- Vectorize(tailq)
}
else {
    z.U <- tailcdf(upper.threshold)
    taild1 <- function(x) {
        as.numeric((do.call(ddistname, c(list(x), as.list(pars)))))
    }
    taild <- function(x) {
        eval <- taild1(x)/(z.U - z.H)
        if (!is.na(eval)) {
            return(eval)
        }
        else return(0)
    }
    taild <- Vectorize(taild)
    tailq1 <- function(x) {
        as.numeric((do.call(qdistname, c(list(x), as.list(pars)))))
    }
    tailq <- function(q) {
        tailq1(q * (z.U - z.H) + z.H)
    }
    tailq <- Vectorize(tailq)
}
z <- tailcdf(data)/z.U
n <- length(data)
rvs <- matrix(runif(n * 5000, min = z.H, max = z.U), nrow = 5000,
    ncol = n)
rvs <- apply(rvs, 1, function(x) {
```

```
    x[order(x, decreasing = FALSE)]
})
KS <- KS.test(z, n, z.H)
KS.pvalue <- 1 - ecdf(apply(rvs, 2, KS.test, n = n, z.H = z.H))(KS)
V <- V.test(z, n, z.H)
V.pvalue <- 1 - ecdf(apply(rvs, 2, V.test, n = n, z.H = z.H))(V)
AD <- AD.test(z, n, z.H)
AD.pvalue <- 1 - ecdf(apply(rvs, 2, AD.test, n = n, z.H = z.H))(AD)
AD.up <- AD.up.test(z, n, z.H)
AD.up.pvalue <- 1 - ecdf(apply(rvs, 2, AD.up.test, n = n,
    z.H = z.H))(AD.up)
AD2 <- AD2.test(z, n, z.H)
AD2.pvalue <- 1 - ecdf(apply(rvs, 2, AD2.test, n = n, z.H = z.H))(AD2)
W2 <- W2.test(z, n, z.H)
W2.pvalue <- 1 - ecdf(apply(rvs, 2, W2.test, n = n, z.H = z.H))(W2)
AD2.up <- AD2.up.test(z, n, z.H)
AD2.up.pvalue <- 1 - ecdf(apply(rvs, 2, AD2.up.test, n = n,
    z.H = z.H))(AD2.up)
Chi2 <- chi2.test(z, n, z.H)
Chi2.pvalue <- 1 - ecdf(apply(rvs, 2, chi2.test, n = n, z.H = z.H))(Chi2)
AD.up.mod1 <- AD.up.mod1.test(z, n, z.H)
AD.up.mod1.test.pvalue <- 1 - ecdf(apply(rvs, 2, AD.up.mod1.test,
    n = n, z.H = z.H))(AD.up.mod1)
value <- matrix(c(KS, V, AD, AD.up, AD2, W2, AD2.up, Chi2,
    AD.up.mod1, KS.pvalue, V.pvalue, AD.pvalue, AD.up.pvalue,
    AD2.pvalue, W2.pvalue, AD2.up.pvalue, Chi2.pvalue, AD.up.mod1.test.pvalue),
    ncol = 2)
rownames(value) <- c("KS", "V", "AD", "AD.up", "AD2", "W2",
    "AD2.up", "Chi2", "AD.up.mod1")
colnames(value) <- c("Statistic", "P-Value")
if (!log.scale) {
    hist(log(data), freq = FALSE, ylim = c(0, 1))
    curve(taild(exp(x)) * exp(x), add = TRUE, col = "red")
    pseq <- (1:(n - 1))/n
    y <- log(tailq(pseq))
    plot(y, log(data[order(data)][-n]), main = "QQ-Plot",
        xlab = "Theoretical Quantile", ylab = "Sample Quantile")
    abline(a = 0, b = 1, col = "red")
}
else {
    hist(data, freq = FALSE, ylim = c(0, 1))
    curve(taild(x), add = TRUE, col = "red")
    pseq <- (1:(n - 1))/n
    y <- tailq(pseq)
    plot(y, data[order(data)][-n], main = "QQ-Plot",
```

```
            xlab = "Theoretical Quantile",
            ylab = "Sample Quantile")
        abline(a = 0, b = 1, col = "red")
    }
    d0 <- tailq(pseq) - data[order(data)][-n]
    vio.ratio <- sum((sign(d0) * (sign(d0) + 1)/2)/length(d0))
    w.vio.ratio <- sum((d0 * (sign(d0) + 1)/2)/sum(abs(d0)))
    value <- rbind(value, matrix(c(vio.ratio, w.vio.ratio, NA,
        NA), ncol = 2))
    n.value <- nrow(value)
    rownames(value)[(n.value - 1):n.value] <- c("Vio.Ratio",
        "W.Vio.Ratio")
    return(value)
}
```

`fitter` *Fit a truncated distribution to a given data sample and assess the fit.*

Description

fitter includes truncated maximum likelihood for a given threshold function AND tools for the analysis of GOF.

Usage

```
fitter(data, threshold, distr, start = NULL,
  fix.arg = NULL, pos = TRUE, method = "mle",
  log.scale = FALSE, upper.threshold = Inf,
  method.pareto = "ml", blocksize = NULL,
  quantiles = NULL, se.out = FALSE, weights = NULL)
```

Arguments

`data`	Positive loss data
`threshold`	Tail threshold for the truncated distribution
`distr`	string of a function with pdistr, qdistr and ddistr function, e.g. "norm"
`start`	Starting value for optimization. For the most common distributions it is not necessary to give a starting value
`fix.arg`	Arguments which should be held fix in the optimization
`pos`	Until now two different forms of optimization are implemented: arbitrary number of parameters, all positive (pos = TRUE) or two parameters, one arbitrary, they other one positive e.g. (pos = c(FALSE,TRUE))

`method` Parameter estimation method: can be one of c(”mle”, ”blocks”, ”QQ”, ”chisq”, ”AD.up”, ”AD2.up”, ”AD”, ”AD.up.mod1”, ”AD.up.mod2”, ”AD2.up.mod1”, ”AD2.up.mod2”). Until now ONLY ”mle”, ”blocks” (fitting the distribution of blockwise maxima) and ”QQ” (quantile matchin algorithm) works.

`log.scale` indicates whether the distribution should be fitted on the logarithm of the data

`upper.threshold` fitting a distribution between two points on the real line

`method.pareto` method for finding the shape of pareto (distr = ”pareto1”). Can be one of c(”ml”, ”Hill”, ”ISE”, ”QQ”, ”PDC”, ”WML”), see documentation of package ”laeken” for more information.

`blocksize` Used for parameter estimation via blockwise maxima.

`quantiles` Used for the quantile matching optimization to find the parameters.

`se.out` If TRUE and method = ”mle”, the standard errors for the optimization will be returned.

`weights` If not NULL, weighted maximum likelihood will be performed.

Value

Analysis of GOF, parameters

Examples

```
body.data <- rlnorm(100,sdlog = 1.5, meanlog = 11)
tail.data <- rlnorm(100,sdlog = 2.5, meanlog = 11)
loss.data <- c(body.data[body.data <= 1e05], tail.data[tail.data > 1e05])
## Not run: mod <- fitter(loss.data,distr = "lnorm", threshold = 1e05)
```

Code

```
function(data,threshold,distr,start= NULL,
         fix.arg = NULL,pos = TRUE,method = "mle",
         log.scale = FALSE,upper.threshold = Inf,
         method.pareto = "ml",blocksize = NULL,
         quantiles = NULL, se.out = FALSE, weights = NULL){

  if(! (method %in% c("mle","blocks", "QQ", "chisq",
                      "AD.up","AD2.up", "AD","AD.up.mod1",
                      "AD.up.mod2", "AD2.up.mod1","AD2.up.mod2"))) {
    stop("Method not known!")
  }

  if(log.scale){
    obs <- log(data)
    obs.t <- log(threshold)
```

```
  obs.ut <- log(upper.threshold)
}else{
  obs <- data
  obs.t <- threshold
  obs.ut <- upper.threshold
}
if(method == "mle"){
mod <- fittruncdistr(data = obs,threshold = obs.t,distr = distr,
                     start = start, fix.arg = fix.arg, pos = pos,
                     upper.threshold = obs.ut,
                     method.pareto = method.pareto, weights = weights)
pars <- as.list(c(mod$estimate,
                  unlist(fix.arg)))
likelihood <- mod$maximum
}
if(method == "blocks"){
    if(!is.infinite(upper.threshold))
      stop("No Upper Threshold possible in Block
                                            Maxima ML")
    if(is.null(blocksize)) stop("Please define blocksize")
    NC = blocksize
    tmp = matrix(
      obs[obs>=obs.t][1:(NC*floor(length(obs[obs>=obs.t])/NC))],
      nrow = NC)
    NR = dim(tmp)[1]
    mm = apply(tmp, 2, max)
    mod <- fittruncmax(data = mm,
                       threshold = obs.t,distr = distr,
                       mlen = NR,start = start,
                       fix.arg = fix.arg, pos = pos,
                       method.pareto = method.pareto)
    pars <- as.list(c(mod$estimate,
                      unlist(fix.arg)))
    likelihood <- mod$maximum
  }

if(method == "QQ" | method == "chisq" |
     method == "AD.up" | method == "AD" |
     method == "AD2.up" | method == "AD2.up.mod2" |
     method == "AD2.up.mod1" |
     method == "AD.up.mod1" | method == "AD.up.mod2"){
  if(!is.infinite(upper.threshold))
    stop("No Upper Threshold possible with this method")
  if(method == "QQ") if(is.null(quantiles))
    stop("Please define quantiles")
```

```
  mod <- matchqtruncdistr(data = obs, threshold = obs.t,
                          distr = distr,method = method,
                          quantiles = quantiles, start = start,
                          fix.arg = fix.arg)
  pars <- as.list(c(mod$estimate,
                    unlist(fix.arg)))
}


if(is.finite(upper.threshold)){
  data <- data[data <= upper.threshold]}
stats <- gof_tail_for_fitter(data = data,threshold = threshold,
                             distr = distr,pars,log.scale = log.scale,
                             upper.threshold = upper.threshold)

if(method == "mle" | method == "blocks"){
n.data <- length(data>=threshold)
BIC <- -2*likelihood + length(mod$estimate)*log(n.data)
AIC <- 2*(length(mod$estimate) - likelihood)
ICs <- c(AIC = AIC, BIC = BIC)
LLH <- c(LogLH = likelihood)
if(se.out == TRUE){  hessian <- mod$hessian
                          fisher_info<-solve(-hessian)
                          prop_sigma<-sqrt(diag(fisher_info))
                          prop_sigma<-diag(prop_sigma)
                           prop_sigma<-c(diag(prop_sigma))
                          upper<-unlist(mod$estimate)+1.96*prop_sigma
                          lower<-unlist(mod$estimate)-1.96*prop_sigma
                         return( list(Parameters = unlist(pars),
                                      Stats = stats, IC = ICs,
                                      LLH = LLH,
                         interval.lower = c(lower),
                                      interval.upper = c(upper)))}

return( list(Parameters = unlist(pars), Stats = stats,
             IC = ICs, LLH = LLH) )
}else{
  if(method == "QQ"){
 return(list( Parameters = unlist(pars), Stats = stats,
              Empq = mod$empq, Quantiles = mod$quantiles))
  }else{

    return(list( Parameters = unlist(pars), Stats = stats,
                 Method = method, Value = mod$value))
```

```
    }
  }

}
```

6.1.2 Determining the OpVaR

`compquantile` *Computation of a quantile of a compound Poisson distribution via FFT*

Description

Given a compound Poisson distribution, i.e. a random sum of a given distribution function f, this function uses FFT to compute the quantile in a very efficient and fast way.

Usage

```
compquantile(q, f, ..., lambda, h = 0.01, lim = 500,
  intervals = FALSE)
```

Arguments

`q`	Quantile (e.g. 0.999)
`f`	Distribution function
`...`	Parameters of f
`lambda`	Parameter lambda of the Poisson RV
`h`	Stepsize of discretization for FFT
`lim`	Max. limit of discretization
`intervals`	If true, numeric error intervals are given.

Value

Quantile and the error intervals

Examples

```
compquantile(0.999, plnorm, meanlog = 11, sdlog = 1.5, lambda = 20, h = 1e06, lim = 1e08,
intervals = TRUE)
```

Code

```
function (q, f, ..., lambda, h = 0.01, lim = 500, intervals = FALSE)
{
    if (!intervals) {
        return(compquantiletilt(q, f, ..., lambda = lambda, h = h,
            lim = lim, method = "rounding"))
    }
    else {
        qs <- numeric(3)
        qs[1] <- compquantiletilt(q, f, ..., lambda = lambda,
            h = h, lim = lim, method = "upper")
        qs[2] <- compquantiletilt(q, f, ..., lambda = lambda,
            h = h, lim = lim, method = "rounding")
        qs[3] <- compquantiletilt(q, f, ..., lambda = lambda,
            h = h, lim = lim, method = "lower")
        names(qs) <- c("lower", "middle", "upper")
        return(qs)
    }
}


function (q, f, ..., lambda, h = 0.01, lim = 500, method = "rounding")
{
    M <- 2^(ceiling(log(lim/h, base = 2)))
    theta <- 20/M
    cdf <- function(x) f(x, ...)
    if (method == "lower")
        corr <- 1
    else corr <- 0
    disc <- discretize(cdf, from = 0, to = (M - corr) * h, by = h,
        method = method)
    disc <- exp(-theta * (0:(M - 1))) * disc
    fhat <- fft(disc, inverse = FALSE)
    P <- exp(lambda * (fhat - 1))
    g <- cumsum(exp(theta * (0:(M - 1))) * Re(1/M * fft(P, inverse = TRUE)))
    if (g[M] >= q) {
        i <- (min(which(g >= q)) - 1) * h
    }
    else {
        stop("Limit is too low!")
    }
    return(i)
}
```

Bibliography

[AD52] Theodore W Anderson and Donald A Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, 23(2):193–212, 1952.

[AD54] Theodore W Anderson and Donald A Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, 1954.

[AK06] Falko Aue and Michael Kalkbrener. LDA at work: Deutsche Bank's approach to quantifying operational risk. *Journal of Operational Risk*, 1(4):49–93, 2006.

[AT13] Andreas Alfons and Matthias Templ. Estimation of social exclusion indicators from complex surveys: The R package laeken. *Journal of Statistical Software*, 54(15):1–25, 2013.

[Bal97] Vijay Balasubramanian. Statistical inference, occam's razor, and statistical mechanics on the space of probability distributions. *Neural computation*, 9(2):349–368, 1997.

[BK] Klaus Böcker and Claudia Klüppelberg. Multivariate models for operational risk.

[Can33] Francesco Paolo Cantelli. Sulla determinazione empirica delle leggi di probabilita. *Giorn. Ist. Ital. Attuari*, 4(421-424), 1933.

[Cop11] Eric W Cope. Penalized likelihood estimators for truncated data. *Journal of Statistical Planning and Inference*, 141(1):345–358, 2011.

[Cra28] Harald Cramér. On the composition of elementary errors. *Scandinavian Actuarial Journal*, 1928(1):13–74 and 141–180, 1928.

[CRF05] Anna Chernobai, Svetlozar Rachev, and Frank Fabozzi. Composite goodness-of-fit tests for left-truncated loss samples. *Department of Statistics and Applied Probability, University of California Santa Barbara*, 2005.

[DdHPdV01] Jon Danielsson, Laurens de Haan, Liang Peng, and Casper G de Vries. Using a bootstrap method to choose the sample fraction in tail index estimation. *Journal of Multivariate analysis*, 76(2):226–248, 2001.

[DGP08] Christophe Dutang, Vincent Goulet, and Mathieu Pigeon. actuar: An R package for actuarial science. *Journal of Statistical Software*, 25(7):38, 2008.

[EF09] Paul Embrechts and Marco Frei. Panjer recursion versus FFT for compound distributions. *Mathematical Methods of Operations Research*, 69(3):497–508, 2009.

[ELM03] Paul Embrechts, Filip Lindskog, and Alexander McNeil. Modelling dependence with copulas and applications to risk management. *Handbook of heavy tailed distributions in finance*, 8(1):329–384, 2003.

[FdHP03] Ana Ferreira, L de Haan, and Liang Peng. On optimising the estimation of high quantiles of a probability distribution. *Statistics*, 37(5):401–434, 2003.

[Fré51] Maurice Fréchet. Sur les tableaux de corrélation dont les marges sont données. *Ann. Univ. Lyon Sect. A*, 9:53–77, 1951.

[G+91] John Geweke et al. *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Federal Reserve Bank of Minneapolis, Research Department, 1991.

[GB09] Alan Genz and Frank Bretz. *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer-Verlag, Heidelberg, 2009.

[Ger79] Hans U Gerber. *An introduction to mathematical risk theory*, volume 8. SS Huebner Foundation for Insurance Education, Wharton School, University of Pennsylvania Philadelphia, 1979.

[GF07] Christian Genest and Anne-Catherine Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering*, 12(4):347–368, 2007.

[GH99] Rudolf Grübel and Renate Hermesmeier. Computation of compound distributions i: Aliasing errors and exponential tilting. *Astin Bulletin*, 29(2):197–214, 1999.

[GO01] M Ivette Gomes and Orlando Oliveira. The bootstrap methodology in statistics of extremes-choice of the optimal sample fraction. *Extremes*, 4(4):331–358, 2001.

[GRG96] A Gelman, G Roberts, and W Gilks. Efficient metropolis jumping rules. *Bayesian statistics*, 5:599–608, 1996.

[Han82] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054, 1982.

[HT11] Arne Henningsen and Ott Toomet. maxlik: A package for maximum likelihood estimation in R. *Computational Statistics*, 26(3):443–458, 2011.

[HW83] Philip Heidelberger and Peter D Welch. Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6):1109–1144, 1983.

[Jef61] Harold Jeffrey. Theory of probability. *Clarendon Press, Oxford*, page 72, 1961.

[Kol33] Andrey N Kolmogorov. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4(1):83–91, 1933.

[LR13] Vincent Lehérissé and Alexis Renaudin. Quantile distance estimation for operational risk: a practical application. *Journal of Operational Risk Volume*, 8(2):73–102, 2013.

[MFE10] Alexander J McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative risk management: concepts, techniques, and tools*. Princeton university press, 2010.

[Mik09] Thomas Mikosch. *Non-life insurance mathematics: an introduction with the Poisson process*. Springer, 2009.

[NECD06] Johanna Nešlehová, Paul Embrechts, and Valérie Chavez-Demoulin. Infinite mean models and the LDA for operational risk. *Journal of Operational Risk*, 1(1):3–25, 2006.

[NG06] Oesterreichische Nationalbank and Finanzmarkt Austria Dienstleistungs GesmbH. *Guidelines on operational risk management*. OeNB and FMA, 2006.

[oBS03] Basel Committee on Banking Supervision. *The 2002 Loss Data Collection Exercise for Operational Risk: Summary of the Data Collected*. Bank for International Settlement, Basel, Switzerland, March 2003.

[oBS05] Basel Committee on Banking Supervision. *Basel II: International convergence of capital measurement and capital standards - A revised framework*. Bank for International Settlement, Basel, Switzerland, November 2005.

[oBS09] Basel Committee on Banking Supervision. *Observed range of practice in key elements of Advanced Measurement Approaches (AMA)*. Bank for International Settlement, Basel, Switzerland, July 2009.

[oBS11] Basel Committee on Banking Supervision. *Operational Risk - Supervisory Guidelines for the Advanced Measurement Approaches*. Bank for International Settlement, Basel, Switzerland, June 2011.

[Pan81] Harry H Panjer. Recursive evaluation of a family of compound distributions. *Astin Bulletin*, 12(1):22–26, 1981.

[PBCV06] Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. Coda: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11, 2006.

[Roe02] Ancus Roehr. Modelling operational losses. *Algo research quarterly*, 5(2):53–64, 2002.

[Sha06] JEH Shaw. Some quotable quotes for statistics, 2006.

[She11] Pavel V Shevchenko. *Modelling operational risk using Bayesian inference*. Springer, 2011.

[Skl59] M Sklar. *Fonctions de répartition à n dimensions et leurs marges*. Université Paris 8, 1959.

[SM12] Carl Scarrott and Anna MacDonald. A review of extreme value threshold estimation and uncertainty quantification. *REVSTAT–Statistical Journal*, 10(1):33–60, 2012.

[Smi48] Nickolay Smirnov. Table for estimating the goodness of fit of empirical distributions. *The Annals of Mathematical Statistics*, 19(2):279–281, 1948.

[Ste74] Michael A Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.

[Str97] Erwin Straub. *Non-life insurance mathematics*. Springer, 1997.

[VM28] Richard Von Mises. *Wahrscheinlichkeit, Statistik und Wahrheit*. Berlin, 1928.

[Wal44] Abraham Wald. On cumulative sums of random variables. *The Annals of Mathematical Statistics*, 15(3):283–296, 1944.

[ZGFT13] Xiaoping Zhou, Rosella Giacometti, Frank J Fabozzi, and Ann H Tucker. Bayesian estimation of truncated data with applications to operational risk measurement. *Quantitative Finance*, (ahead-of-print):1–26, 2013.