Report I - MT7027 Risk Models and Reserving in Non-Life Insurance

Michael Ståhle
880917-0437

Robert Holmlund
850605-0239

March 2017

Project I

Objectives

In this project we will fit a counting process to claim arrivals and model the claim size distributions for two insurance products. The data includes daily observations in two different non-life insurance branches, on the number of claims and the claim sizes over 10 years. The projects goal is to model the distribution of the total claim cost for the entire company. This will be done by Monte Carlo simulation and historical sampling. We will then study the behaviour of the total claim cost distribution under different reinsurance arrangements such as XL- and SL-covers. The necessary assumptions for the simulation will be stated and a theoretical section is provided. The produced results visualize the altering behavior of the claim cost distribution and it's quantile function.

Mathematical Background

The Poisson Process

The Poisson Process will be of major importance in the coming analysis. For that reason we will reproduce the rigorous definition that is given in the book Non-Life Insurance Mathematics by Thomas Mikosh. [1] It is necessary to introduce some notation. For any real-valued function f on $[0, \infty)$ we write

$$f(s,t] = f(t) - f(s), \qquad 0 \leq s < t < \infty.$$

Define an integer-valued random variable M, then M is said to have a Poisson distribution with parameter $\lambda > 0$ if it has distribution

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

If M is Poisson distributed with parameter λ we write $M \sim Pois(\lambda)$. We can now define the Poisson process.

Definition 1 (Poisson Process)
A stochastic process $N = (N(t))_{t \geq 0}$ is said to be a Poisson process if the following conditions hold:

1. The process starts at zero: $N(0) = 0$ $a.s.$

2. The process has independent increments: for any t_i, $i = 0, ..., n$ and $n \geq 1$ such that $0 = t_0 < t_1 < ... < t_n$, the increments $N(t_{i-1}, t_i]$, $i = 1, ..., n$, are mutually independent.

3. There exists a non-decreasing right-continues function $\mu : [0,\infty] \to [0,\infty)$ with $\mu(0) = 0$ such that increments $N(s,t]$ for $0 < s < t < \infty$ have a Poisson distribution $Pois(\mu[s,t))$. We call μ the mean function of N.
4. With probability 1, the sample paths $(N(t,\omega))_{t\geq 0}$ of the process N are right continues for $t \geq 0$ and have limits from the left for $t > 0$. We say that N has càdlàg (continue à droite, limites à gauche) sample paths.

In the context of insurance data we view the claim arrival process as a Poisson process. From point 3 in the definition we have that

$$N(t) = N(t) - N(0) = N(0,t] \sim Pois(\mu(t))$$

i.e. the number of insurance claims in the time interval $[0,t]$ is assumed to be Poisson distributed with mean function $\mu(t)$. If the mean function is linear in t:

$$\mu(t) = \lambda t \quad t \geq 0,\ \lambda > 0$$

the process is called homogeneous, otherwise it's called inhomogeneous. Generally the mean value function has the representation:

$$\mu(s,t] = \int_s^t \lambda(y)dy.$$

Where $\lambda(y)$ is called the intensity function and $\mu(s,t] = \mu(t)$ when $s = 0$. $\mu(s,t]$ can be interpreted as the inner clock of the Poisson process. The inhomogeneous process $N(t)$ "slows down" or "speeds up" depending on the magnitude of $\lambda(t)$. This is a convenient way of modelling seasonal trends in insurance data. One example is that the number of car accidents increase during the winter because of icy roads.

A common situation in the insurance business is when λ is stochastic, extra cold winters generates very icy roads and many accidents. Let Λ be a random variable with density function g and $\tilde{N}(t)$ be a counting process, then $\tilde{N}(t)$ is called a mixed poisson process if $(\tilde{N}(t)|\Lambda = \lambda) \sim Pois(\lambda t)$. [2] We have that:

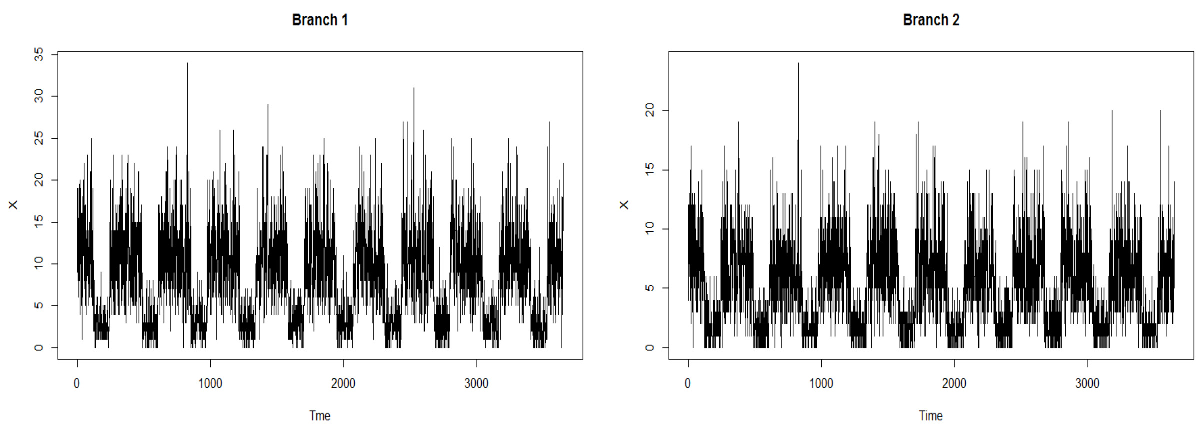
$$P(\tilde{N}(t) = k) = \int_0^\infty e^{\lambda t}\frac{(\lambda t)^k}{k!}g(\lambda)d\lambda$$

.

Claim arrival process

Let $X_i = (x_1, x_2, ..., x_{3650})$ for $i = 1,2$ represent the data of claim arrivals for branch i. We have that x_j is the number of claims day j, note that we have 3650 days which is consistent with 10 years. A plot of X_i with time $T = (1, ..., 3650)$ on the x-axis reveals a periodic behavior of the claim arrivals.

Figure 1: Number of claim arrivals per day



According to Figure 1 the claim arrivals have periods with a higher arrival rate and periods with a lower arrival rate. If we assume that our data starts in January and consider the number of claims per month, then it becomes clear that the high rate periods are September to April and the low rate periods are May to August.

The goal is to fit a model to the claim arrival process in order to predict future claims. A good suggestion for the model is the Poisson process. Since the data does not include interarrival times we can't directly fit a Poisson process. Instead we consider blocks of arrivals and assume they are Poisson distributed with rate $t\lambda$, where t is the number of time blocks. First we consider the time blocks as days.

From Figure 1 we concluded that the arrival process has a periodic behavior. Let periods with higher intensity have rate λ_h and periods with lower intensity have rate λ_l, then we define the inhomogeneous Poisson process $N(\mu(t))$ where $\mu(t) = \lambda_i t$ and $i \in (h, l)$ i.e. the process has a linear mean value function given the intensity period. By the third condition in Definition 1 we have that:

$$P(N(t\lambda_i) = k) = e^{-t\lambda_i} \frac{(t\lambda_i)^k}{k!}.$$

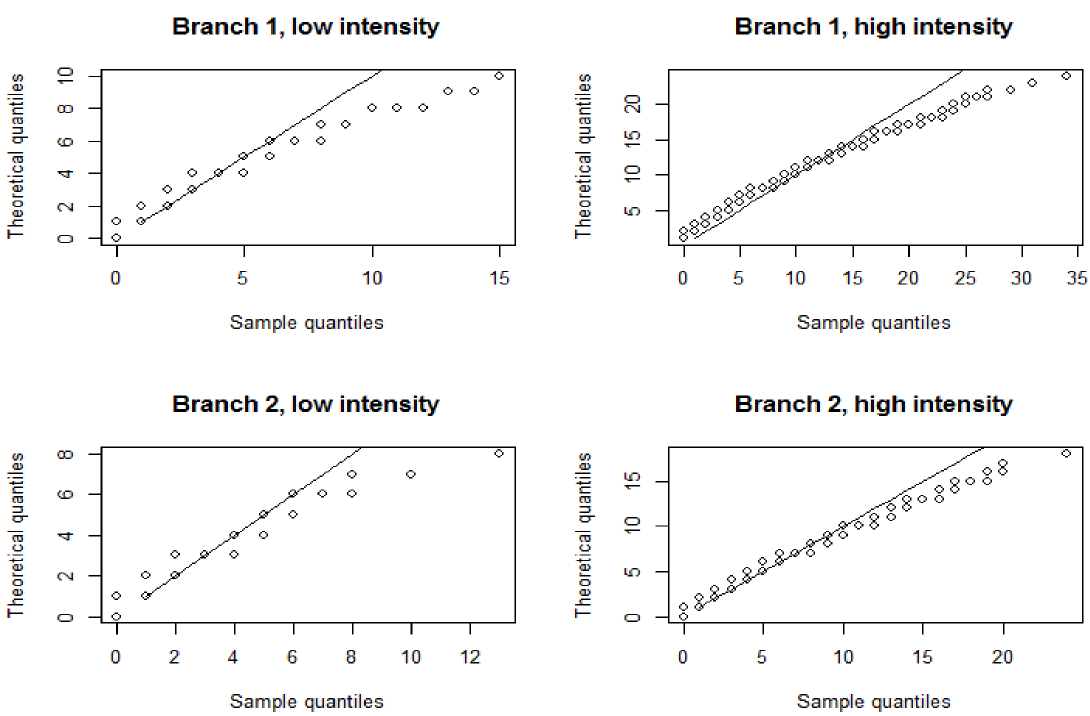
The maximum likelihood estimate of λ_i equals the sample mean of the claims. [3] The estimated intensities are presented in Table 1.

Table 1: My caption

	Branch 1	Branch 2
$\hat{\lambda}_h$	10.959	6.863
$\hat{\lambda}_l$	2.831	1.770

To see if the Poisson distribution fits the data we plot the quantiles of the empirical distribution against the quantiles of a Poisson distribution with the parameters in Table 1. The plot is called a qq-plot and Figure 2 shows the resulting plots of the two branches and intensity periods.

Figure 2: qq-plots for the assessment of the poisson distribution assumption.



If the Poisson distribution had a perfect fit, the theoretical quantiles would equal the sample quantiles and all the dots would be on the straight line $y = x$. The four plots in Figure 2 have one common feature, the plotted quantiles departs from the straight lines for higher values. The behavior shows that the sample distributions have fatter right tails compared to the theoretical Poisson distributions. The second condition in Definition 1 says that the Poisson process has independent increments i.e. the number of claims one day should be independent of the claims the day before. In other words should the auto correlation function

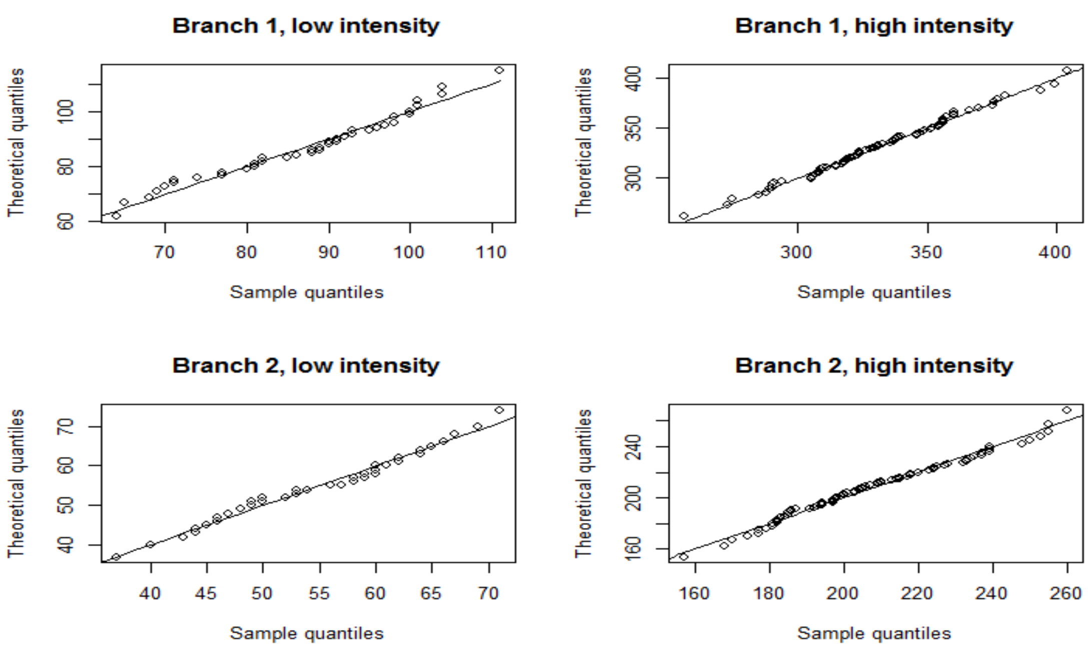
$$\rho_l = corr(x_t, x_{t-l})$$

be zero for all l. ACF plots and Ljung-Box tests (see Ruey S. Tsay Analysis of Financial Time Series for details [4]) suggests that autocorrelation is present for lag 1 i.e. $\rho_1 > 0$. To address the problem with autocorrelation we extend the time blocks to months. Further we need a distribution that has a fatter tail compared to the Poisson distribution. A natural choice is the Negative Binomial distribution which is the special case of a compounded Poisson distribution where $\Lambda \sim Gamma(\alpha, \beta)$. The distribution function for the negative binomial distribution is

$$P_k(\alpha, \beta) = \frac{\Gamma(\alpha + k)}{\Gamma(\alpha)k!}(\frac{\beta}{\beta + 1})^{\alpha}(\frac{1}{\beta + 1})^k$$

and we get the maximum likelihood estimates by numerically solving the ML-equations (see Johansson [3] for derivations). Now we evaluate the goodness of fit by qq-plots presented in Figure 3.

Figure 3: qq-plots for the assessment of the negbin distribution assumption.



The fit is improved, compared to the plots in Figure 2 the plotted quantiles stays on or close to the straight line for higher values. The negative binomial distribution has a satisfying fit to the data.

Claim size distribution

Let $\mathbf{Y}_i = (Y_{1i}, Y_{2i}, ..., Y_{N_i(t)})$ for $i = 1, 2$ represent the random claim sizes for branch i. Note that the total number of claim sizes $N_i(t)$ is defined by the mixed Poisson process derived in the previous section. Denote the sum of the claim sizes for branch i by $S_i(t)$, then the total claim cost per branch for one year is

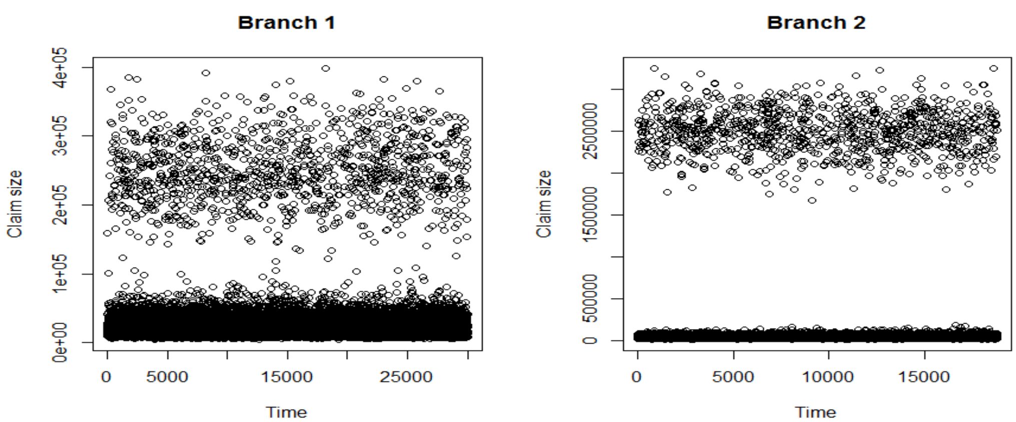
$$S_i(12) = \sum_{k=1}^{N(12)} Y_{ki}. \tag{1}$$

Remember that we consider months as time unit. Given three standard assumptions we say that the distribution for S_i has a compound negative binomial distribution. [5] The assumptions are as follows:

1. N is a discrete random variable which only takes values in $\mathbb{N}_0$.
2. $Y_1, Y_2, ... \sim iid$ with a general distribution function G, $G(0) = 0$.
3. N and $(Y_1, Y_2, ...)$ are independent.

Since $N = N_i(t)$ for branch i is defined as the special case of the mixed Poisson process where N has a negative binomial distribution, assumption one holds. Next we want to check the independence of $(Y_1, Y_2, ...)$ and find their distribution function G. Lets plot the claim sizes.

Figure 4: Plots of claim sizes for Branch 1 and 2



The plots in Figure 4 displays a similar pattern for both branches. The distribution of the claim size seem to be divided in two parts, one for large claims and one for small claims. Define a threshold M and let $G_{lc}(y) = P(Y_1 \leq y | Y_1 > M)$ be the distribution for the large claims and let $G_{sc}(y) = P(Y_1 \leq y | Y_1 < M)$ be the distribution for the small claims. It follows that

$$S_{sc,i} = \sum_{k=1}^{N_i} Y_k * I(Y_k < M)$$

$$S_{lc,i} = \sum_{k=1}^{N_i} Y_k * I(Y_k > M)$$

is also compound negative distributed. By an alternative parametrization of the negative binomial distribution that utilizing the relation with the mixed Poisson distribution, we write S_i in (1) $S_i \sim CompNB(\lambda v, \gamma, G)$. [5] It then follows that

$$S_{lc,i} \sim CompNB(\lambda_i(1 - G_i(M))v_i, \gamma, G_{lc,i})$$

$$S_{sc,i} \sim CompNB(\lambda_i G_i(M) v_i, \gamma, G_{sc,i}).$$

We now try and estimate different common distributions for $G_{lc,i}$ and $G_{sc,i}$ and evaluate their fit by qq-plots. The distributions with the best fit were as follows:

$$G_{lc,1} \sim N(\mu = 249610.5, \sigma = 49019.16)$$

$$G_{sc,1} \sim LN(\mu_{LN} = 9.90768, \sigma_{LN} = 0.42188)$$

$$G_{lc,2} \sim N(\mu = 2496720, \sigma = 255468.1)$$

$$G_{sc,2} \sim LogGamma(Shape = 670.1945, Rate = 64.5249)$$

N stands for the normal distribution and LN stands for the log-normal distribution, qq-plots are presented in Figure 5 and 6. For detailed descriptions of the distributions see Wuthrich. [5]

Figure 5: qq-plots for large claims

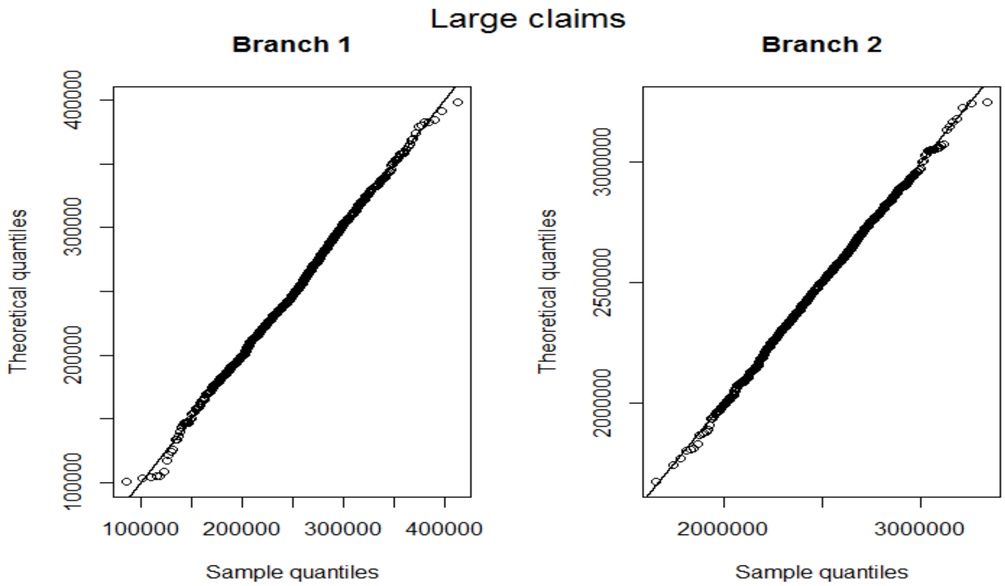
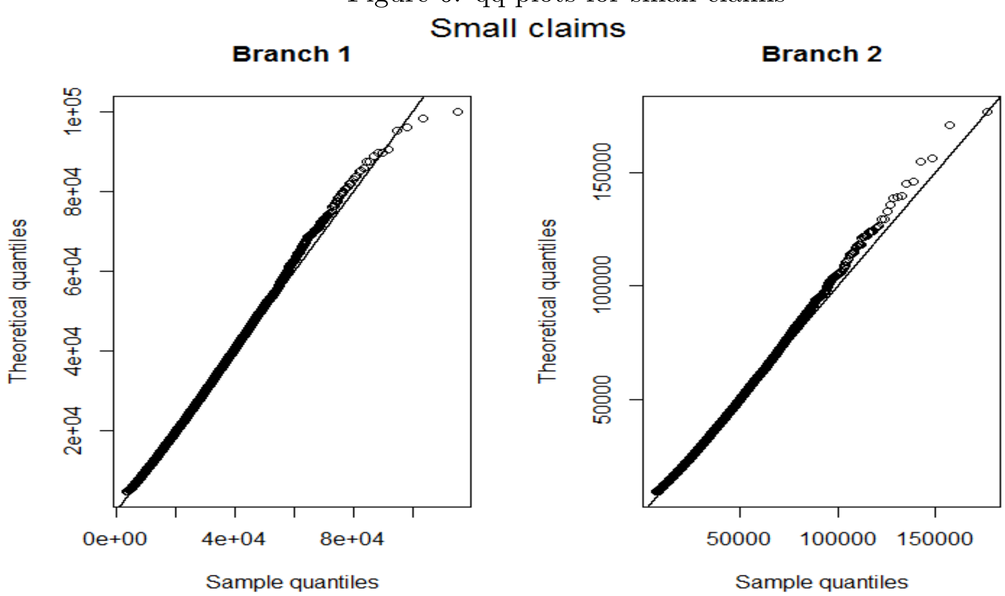


Figure 6: qq-plots for small claims



The qq-plots in Figure 5 shows a satisfying fit of the normal distribution. The qq-plots for small claims in Figure 6 deviates more from the straight line (especially for higher quantiles) but the overall fit is satisfying. Regarding the independence assumption of $(Y_1, Y_2, ...)$, the plots in Figure 4 does not display seasonal behaviour. Furthermore ACF plots and Ljung Box tests shows no signs of autocorrelation. Last we conclude that the independence between $(Y_1, Y_2, ...)$ and N is a fair assumption since the plots in Figure 4 does not have a periodic behavior.

XL/SL covers

Lets expand the stochastic sum in (1)

$$S_i = \sum_{k=1}^{N_i} Y_{ki} = \sum_{k=1}^{N_i} max((Y_{ki} - u_{xl,i}), 0) + \sum_{k=1}^{N_i} min(Y_{ki}, u_{xl,i})$$

where $u_{xl,i}$ is a constant. Then the part the insurance company want to reinsure is

$$\sum_{k=1}^{N_i} max((Y_{ki} - u_{xl,i}), 0)$$

it's called XL-cover (Excess Loss cover). We choose $u_{xl,i}$ to be the 90% quantile of the emperical distribution of claim sizes for branch i. The final price that is payed to the reinsurance company is set to 110% of the expected cost i.e.

$$E[\sum_{k=1}^{N_i} max((Y_{ki} - u_{xl,i}), 0)] * 1.1.$$

Now we look at the decomposition:

$$S_i = \sum_{k=1}^{N_i} Y_{ki} = max((S_i - u_{sl,i}), 0) + min(S_i, u_{sl,i})$$

where $u_{sl,i}$ is a constant. The part that the reinsurance company want to reinsure is

$$max((S_i - u_{sl,i}), 0)$$

it's called SL-cover (Stop Loss cover). We choose $u_{sl,i}$ to be the 90% quantile of the empirical distribution of the total claim sizes and the final price that is payed to the reinsurance company is set to 110% of the expected cost i.e.

$$E[max((S_i - u_{sl,i}), 0)] * 1.1.$$

Monte Carlo simulation

In order to calculate the expected values of the XL- and SL-covers we need the distribution of

$$\sum_{k=1}^{N_i} max((Y_{ki} - u_{xl,i}), 0)$$

and

$$max((S_i - u_{sl,i}), 0).$$

It is a difficult task to derive their distributions theoretically, so we simulate data from the distributions derived in previous sections and calculate the Xl- and SL-covers. We then repeat the process multiple times and use the means as estimates for the true expected values. This procedure is called Monte Carlo simulation. Define the set of values for branch $i = 1, 2$:

$$\{N_i^{(j)}, Y_{1i}^{(j)}, Y_{2i}^{(j)}, ...Y_{N_i^{(j)},i}^{(j)}; j = 1, ..., n\}$$

and create:

$$Z_{SL,i}^{(j)} = max(\sum_{k=1}^{N_i^j} Y_{ki}^{(j)} - u_{SL,i}, 0)$$

and

$$Z_{XL,i}^{(j)} = \sum_{k=1}^{N_i^j} max(Y_{ki}^{(j)} - u_{XL,i}, 0)$$

then by the law of large numbers we have that $E[Z] \to \frac{1}{n}\sum_{j=1}^{n} Z^{(j)}$ when $n \to \infty$. In order for the simulation results to be adequate we assume the following:

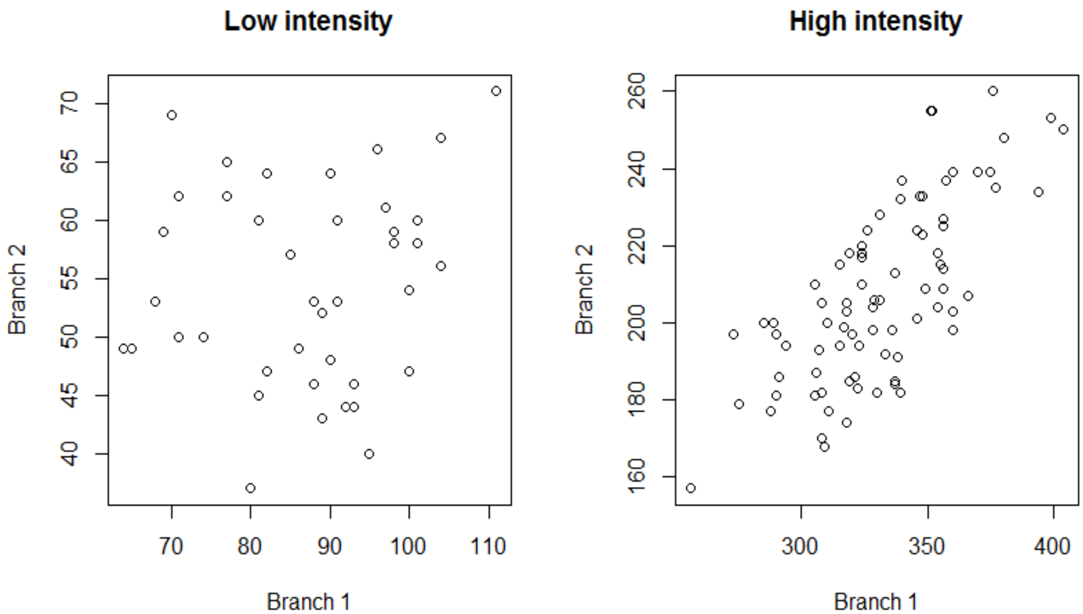
1. $N_i^{(j)}$ and $(Y_{1i}^{(j)}, Y_{2i}^{(j)}, ...Y_{N_i^{(j)},i}^{(j)})$ are independent.

2. $(Y_{1i}^{(j)}, Y_{2i}^{(j)}, ...Y_{N_i^{(j)},i}^{(j)})$ are independent.

3. $N_1^{(j)}$ and $N_2^{(j)}$ are independent.

The first two assumptions have been checked in previous sections. For assumption 3, lets plot the number of claims in branch 1 against the number of claims in branch 2.

Figure 7: Number of claims per month.



There is no obvious dependence between branch 1 and 2 in the low intensity period but a clear linear dependence between the branches in the high intensity period. Because of the dependence we can't simulate the number of claims from the negative binomial distribution that we derived in a previous section. To capture the dependence we use historical sampling from the original data i.e drawing with replacement. Since there is no dependence in the low intensity period it is possible to utilize the fitted negative binomial distribution but the results are based on historical sampling for both intensity periods.

Because the first two assumptions of the Monte Carlo simulation holds, we can utilize the distributions for the claim sizes i.e. $Y_{ki}^{(j)}$ is distributed as follows:

$$Y_{k1}^{(j)} \sim \begin{cases} G_{lc,1} & \text{If } Y_{k1}^{(j)} > M_1 \\ G_{sc,1} & \text{If } Y_{k1}^{(j)} < M_1 \end{cases}, \qquad Y_{k2}^{(j)} \sim \begin{cases} G_{lc,2} & \text{If } Y_{k2}^{(j)} > M_2 \\ G_{sc,2} & \text{If } Y_{k2}^{(j)} < M_2 \end{cases}.$$

Consider the two plots in Figure 4 on page 7 and the choice of M. The goal is to find a clear cut-of point, any value between 500000 and 1500000 would work for branch 2. The choice is not that obvious for branch 1. The problem is to decide which distribution the points in the middle belongs to. Johansson [3] suggests a mix of the large and small claims distributions. How ever because Johanssons solution is too time consuming we have chosen

$M_1 = 100000$ and $M_2 = 500000$.

Results

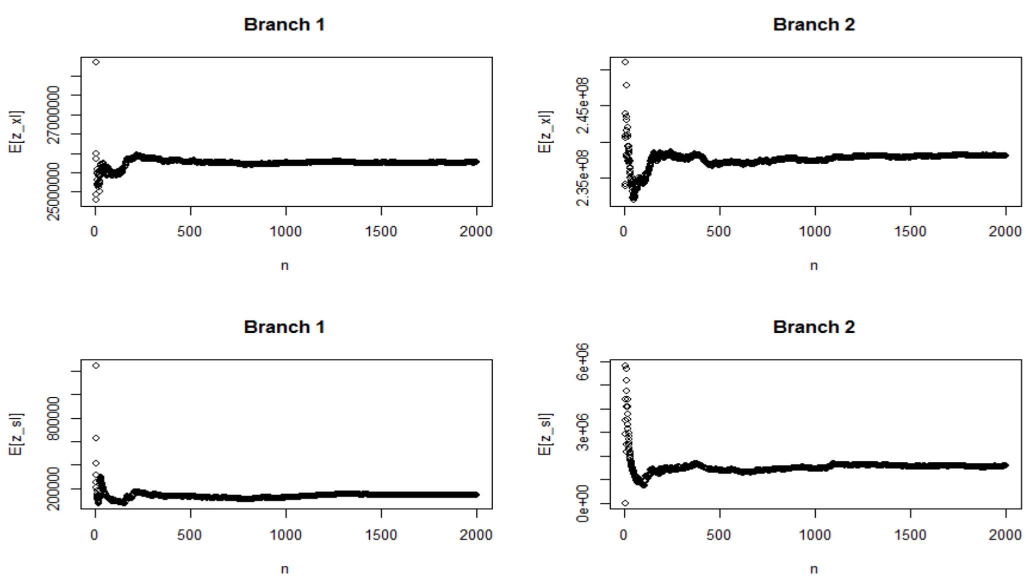
We estimate $u_{xl,i}$ and $u_{sl,i}$ with the 90% quantiles from simulated data and use the values in the simulations for $Z^{(j)}_{SL,i}$ and $Z^{(j)}_{XL,i}$. The result of the simulations are presented in Table 2.

Table 2: Results of the Monte Carlo simulation.

$n = 2000$	Branch 1	Branch 2
u_{xl}	38163.02	63171.34
u_{sl}	96871226	336441115
$E[z_{xl}]$	25762746	238077150
$E[z_{sl}]$	142873.5	1595744
Reinsurance price xl	28339020	261884865
Reinsurance price sl	157160.8	1755318

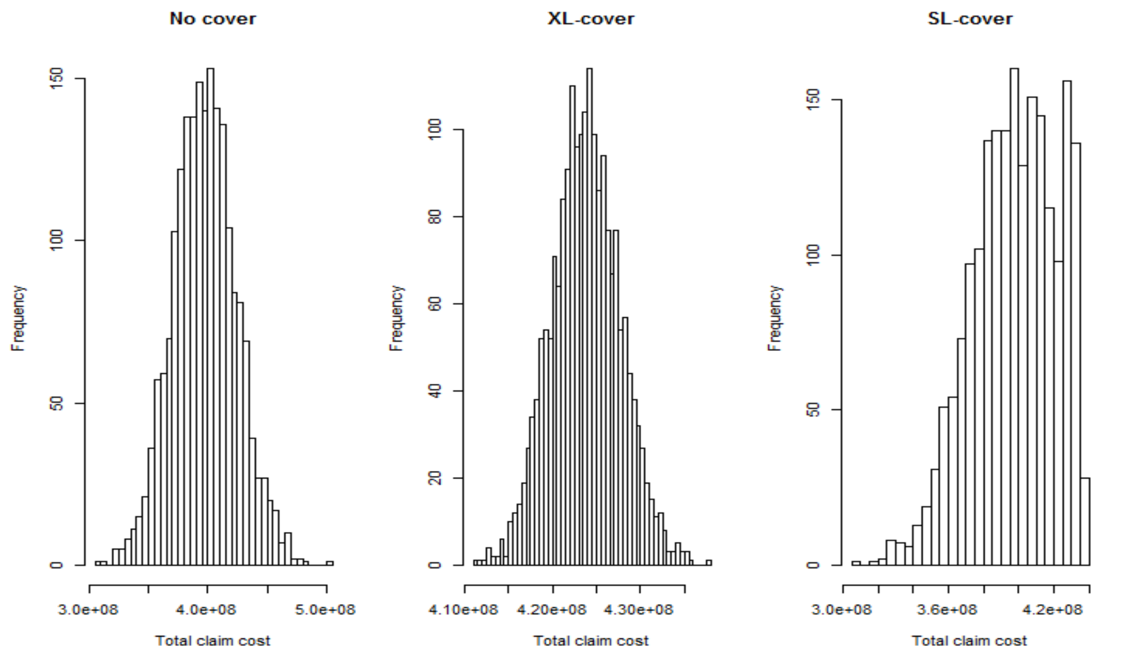
For both branches the mean of the total claim cost from the 2000 simulations is 397402479 which can be compared to the mean of the total cost from historical data 397732239. Further we evaluate the simulation results by plotting the evolution of $E[Z_{SL,i}]$ and $E[Z_{XL,i}]$ when n increases.

Figure 8: Evolution of $E[Z_{SL,i}]$ and $E[Z_{XL,i}]$



The results converges fast, before n=500 in all four cases. The distribution of the total claim cost for the company ($S_1 + S_2$) with the different covers is visualized in Figure 9.

Figure 9: Histograms of the total claim cost.



The histogram for the distribution with no cover is based on the simulated values of S_1+S_2 while the histogram for the distribution with XL-cover is based on the simulated values of:

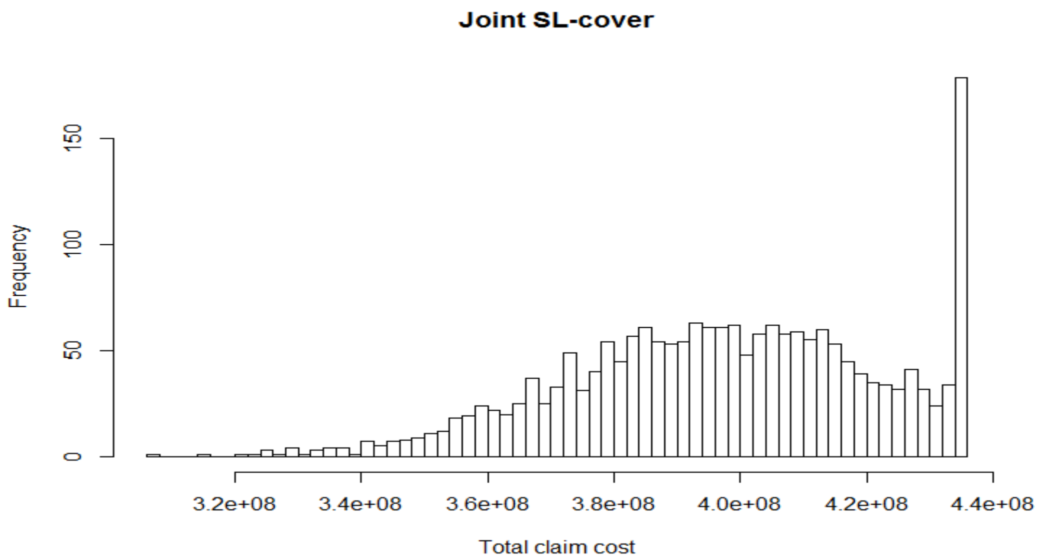
$$\sum_{k=1}^{N_1} min(Y_{k1}, u_{xl,1}) + \sum_{k=1}^{N_2} min(Y_{k2}, u_{xl,2}) + E[Z_{XL,1}] * 1.1 + E[Z_{XL,2}] * 1.1$$

and for SL-cover we have:

$$min(S_1, u_{sl,1}) + min(S_2, u_{sl,2}) + E[Z_{SL,1}] * 1.1 + E[Z_{SL,2}] * 1.1.$$

As expected the distribution with XL-cover is shifted to the right but is more dense compared to the distribution with no cover. The XL-cover successfully decrease probability mass in the right tail of the total claim cost distribution. With SL-cover the total claim cost (including the price of the reinsurance) can't be larger then $1.1 * z_{sl,1} + 1.1 * z_{sl,2} + u_{sl,1} + u_{sl,2} = 435224820$ thus the distribution with SL-cover has a truncated form. The maximum value is then the highest quantile compared to the distribution with XL-cover that theoretically has infinite high quantiles. Next we consider the total claim cost distribution with a joint SL-cover, the distribution is visualized in Figure 10 and the differences in quantile behaviour between the three covers is presented in Figure 11.

Figure 10: Histogram of the total claim cost distribution with joint SL-cover.



The histogram for the distribution with joint SL-cover is based on the simulated values of:

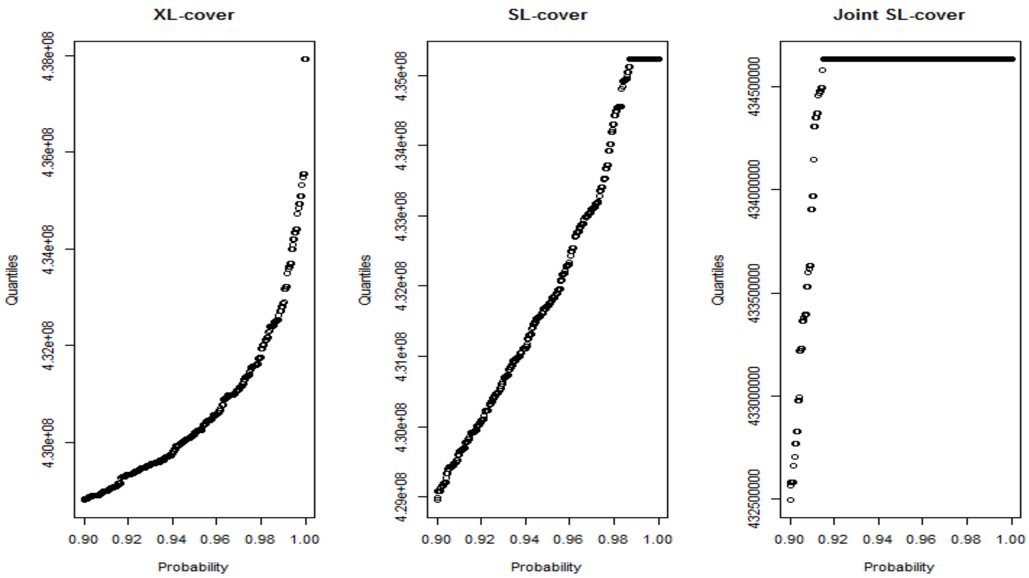
$$min(S_1 + S_2, u_{sl,1} + u_{sl,2}) + max(S_1 + S_2 - (u_{sl,1} + u_{sl,2}), 0) * 1.1$$

where

$$Z_{sl,jointly} = max(S_1 + S_2 - (u_{sl,1} + u_{sl,2}), 0)$$

From Figure 10 we conclude that the joint SL-cover provides even more probability mass in the right end which means that it's quantile function reaches it's maximum faster. This is clear in Figure 11 where the joint SL-cover has a much steeper slope then the mixed SL-cover (middle picture).

Figure 11: Histogram of the total claim cost distribution.



The quantile function for the distribution with XL-cover has a exponential behaviour where it grows slower in the beginning and faster in the end. We note that XL-cover provides lower quantile-values compared to SL-covers for probabilities under 0.99 but higher over 0.99.

Summary

We have found that the claim arrival process has two intensity periods (high and low). Within the periods the process follows the special case of the mixed Poisson process where the number of claims at time t is negative binomial distributed. Autocorrelation at lag 1 is present in the daily observations of claim arrivals, it is successfully removed when modeling the claim arrivals on monthly bases. Claim arrivals are independent in low intensity periods among branches but dependent in high intensity periods.

The total claim cost can be modeled as a compound distribution i.e the stochastic sum of the claim sizes. We conclude that the claim sizes are an independent sequence of stochastic variables and independent from the arrival process. The distribution of claim sizes is split in to two distributions, one for large claims and one for small claims.

Because of the dependence between branches in high intensity periods we use historical resampling in our Monte Carlo simulation. We simulate the distributions for the total claim cost both without and with XL and SL covers. Finally we simulate the total claim cost with a joint SL-cover. We found that the Xl-cover makes the total claim cost distribution more compact. The distributions with SL-cover has a truncated form. Last we have that the quantile function for the distribution with XL-cover has lower values (and increases slower) in the range $(0.9, 0.99)$ then the quantile functions for the SL-covers. How ever the quantile function for XL-cover have theoretically infinite values while the quantile function for SL-covers has a max.

References

[1] Thomas Mikosch, Non-Life Insurance Mathematics *An Introduction with the Poisson Process*, Springer Berlin Heidelberg 2009.

[2] Ross, S.M., Introduction to Probability Models Academic Press, 11th edition, 2014.

[3] Björn Johansson, Matematiska modeller inom sakförsäkring *Kompendium*, Februari 1997 - Nytryck 2014.

[4] Ruey S. Tsay, Analysis of Financial Time Series, 3rd Edition 2010.

[5] Mario V. Wuthrich, Non-Life Insurance: Mathematics and Statistics 2016

Appendix

```
mydata = read.table("~/Riskmodeller, försäkring/Projekt1_Grupp6.txt", header = TRUE, sep =

library('MASS')
library('xts')
library('zoo')
library('dplyr')
branch1 <- filter(mydata, ClaimType==1)
branch2 <- filter(mydata, ClaimType==2)

## Antal claims per dag
numberclaims <- c()
for(i in 1:max(branch1$ClaimDay)){
  numberclaims <- c(numberclaims,length(which(branch1$ClaimDay == i)))
}


## skapar en tidsserie (xts-objekt) för antalet claims
day <- c(1:3650)
myxts <- xts(numberclaims, as.Date(day))
```

```
## Plockar ut alla vinter-observationer (hög intensitet), konverteras om från xts till inte
winterm <- c()
for(i in as.character(c(0:9))){
x1 <- c(myxts[paste0(':197',i,'-01'),],myxts[paste0(':197',i,'-02'),],myxts[paste0(':197',i
winterm <- c(winterm,x1)
}

rm(i)
rm(x1)

hist(winterm,30)
mean(winterm)
var(winterm)

## Plockar ut alla sommar-observationer (låg intensitet), konverteras om från xts till inte
summerm <- c()
for(i in as.character(c(0:9))){
  x1 <- c(myxts[paste0(':197',i,'-05'),],myxts[paste0(':197',i,'-06'),],myxts[paste0(':197'
  summerm <- c(summerm,x1)
}

rm(i)
rm(x1)


## summer branch 1
sort_summerm <- sort(summerm)
qqpois_s <- qpois(ppoints(sort_summerm),mean(summerm))
plot(sort_summerm,qqpois_s)
lines(seq(0:15),seq(0:15))

## winter branch 1
sort_winterm <- sort(winterm)
qqpois_w <- qpois(ppoints(sort_winterm),mean(winterm))
plot(sort_winterm,qqpois_w)
lines(seq(0:30),seq(0:30))

## summer branch 2
sort_summerm2 <- sort(summerm2)
```

```
qqpois_s2 <- qpois(ppoints(sort_summerm2),mean(summerm2))
plot(sort_summerm2,qqpois_s2)
lines(seq(0:10),seq(0:10))


## winter branch 2
sort_winterm2 <- sort(winterm2)
qqpois_w2 <- qpois(ppoints(sort_winterm2),mean(winterm2))
plot(sort_winterm2,qqpois_w2)
lines(seq(0:30),seq(0:30))

### Tidsberoende###
acf(summerm)
acf(winterm)

Box.test(summerm)
Box.test(winterm)

### Gör om till månader

numberclaims <- c()
for(i in 1:max(branch1$ClaimDay)){
  numberclaims <- c(numberclaims,length(which(branch1$ClaimDay == i)))
}
rm(i)

numberclaims2 <- c()
for(i in 1:max(branch2$ClaimDay)){
  numberclaims2 <- c(numberclaims2,length(which(branch2$ClaimDay == i)))
}
rm(i)

day <- c(1:3650)
myxts <- xts(numberclaims, as.Date(day))
myxts2 <- xts(numberclaims2, as.Date(day, origin='1970-01-01'))

claim_m <- apply.monthly(myxts,FUN = sum) ## Skapar antalet claims per månad
claim_m2 <- apply.monthly(myxts2,FUN = sum) ## Skapar antalet claims per månad


wintermm <- c()
```

```
for(i in as.character(c(0:9))){
  x1 <- c(claim_m[paste0(':197',i,'-01'),],claim_m[paste0(':197',i,'-02'),],claim_m[paste0(
  wintermm <- c(wintermm,x1)
}
rm(i)
rm(x1)

summermm <- c()
for(i in as.character(c(0:9))){
  x1 <- c(claim_m[paste0(':197',i,'-05'),],claim_m[paste0(':197',i,'-06'),],claim_m[paste0(
  summermm <- c(summermm,x1)
}
rm(i)
rm(x1)

wintermm2 <- c()
for(i in as.character(c(0:9))){
  x1 <- c(claim_m2[paste0(':197',i,'-01'),],claim_m2[paste0(':197',i,'-02'),],claim_m2[past
  wintermm2 <- c(wintermm2,x1)
}
rm(i)
rm(x1)


summermm2 <- c()
for(i in as.character(c(0:9))){
  x1 <- c(claim_m2[paste0(':197',i,'-05'),],claim_m2[paste0(':197',i,'-06'),],claim_m2[past
  summermm2 <- c(summermm2,x1)
}

rm(i)
rm(x1)


### Tidsberoende###
acf(summermm)
acf(wintermm)

Box.test(summermm)
Box.test(wintermm)
```

```
## Negbin qq

coef_s1 <- fitdistr(summermm,'negative binomial')
coef_w1 <- fitdistr(wintermm, 'negative binomial')
coef_s2 <- fitdistr(summermm2, 'negative binomial')
coef_w2 <- fitdistr(wintermm2,'negative binomial')

## summer branch 1
sort_summermm <- sort(summermm)
qqnegbin_s <- qnbinom(ppoints(sort_summermm),size = coef_s1$estimate[1],mu=coef_s1$estimate
plot(sort_summermm,qqnegbin_s)
lines(seq(0:120),seq(0:120))

## winter branch 1
sort_wintermm <- sort(wintermm)
qqnegbin_w <- qnbinom(ppoints(sort_wintermm),size = coef_w1$estimate[1],mu=coef_w1$estimate
plot(sort_wintermm,qqnegbin_w)
lines(seq(0:400),seq(0:400))

## summer branch 2
sort_summermm2 <- sort(summermm2)
qqnegbin_s2 <- qnbinom(ppoints(sort_summermm2),size = coef_s2$estimate[1],mu=coef_s2$estima
plot(sort_summermm2,qqnegbin_s2)
lines(seq(0:80),seq(0:80))


## winter branch 2
sort_wintermm2 <- sort(wintermm2)
qqnegbin_w2<- qnbinom(ppoints(sort_wintermm2),size = coef_w2$estimate[1],mu=coef_w2$estimat
plot(sort_wintermm2,qqnegbin_w2)
lines(seq(0:270),seq(0:270))

## claim size
plot(branch1$claimcost)
plot(branch2$claimcost)

## qq-plottar claim size
qq_snorm1 <- qnorm(ppoints(branch1_stora), mean = mean(branch1_stora), sd = sd(branch1_stor
sort_branch1s <- sort(branch1_stora)
plot(qq_snorm1,sort_branch1s, ylab = 'Theoretical quantiles', xlab = 'Sample quantiles', ma
lines(seq(0:420000),seq(0:420000))
```

```
b1_sma_lnfit <- fitdistr(branch1_sma,'lognormal')
qq_ln <- qlnorm(ppoints(branch1_sma),b1_sma_lnfit$estimate[1],b1_sma_lnfit$estimate[2])
qqplot(qq_ln,branch1_sma)
lines(seq(0:120000),seq(0:120000))


sort_branch2_stora <- sort(branch2_stora)
qq_branch2_stora <- qnorm(ppoints(branch2_stora),mean = mean(branch2_stora), sd = sd(branch
plot(qq_branch2_stora,sort_branch2_stora, ylab = 'Theoretical quantiles', xlab = 'Sample qu
lines(seq(0:3500000),seq(0:3500000))


log_data <- log(branch2_sma)
b2_sma_lgamma <- fitdistr(log_data, 'gamma')
qq_b2sma_lgamma <- qgamma(ppoints(branch2_sma),shape = b2_sma_lgamma$estimate[1], rate = b2
qqplot(exp(qq_b2sma_lgamma),branch2_sma)
lines(seq(1:200000),seq(1:200000))

sort_b2sma <- sort(branch2_sma)
plot(exp(qq_b2sma_lgamma),sort_b2sma, ylab = 'Theoretical quantiles', xlab = 'Sample quanti
lines(seq(0:200000),seq(0:200000))


## Simuleringsprogram

## simulering
l1 <- 0
n <- 2000
k_xl <- 38163.02
k_xl2 <- 63171.34
k_sl <- 96871226
k_sl2 <- 336441115
z_xl <- c()
z_xl2 <- c()
z_sl <- c()
z_sl2 <- c()
meanz_sl <- c()
meanz_sl2 <- c()
sumskador <- c()
sumskador2 <- c()
```

```
meanz_xl <- c()
meanz_xl2 <- c()
sum_min_yu1 <- c()
sum_min_yu2 <- c()
while(l1 < n){
  a <- round(runif(4,1,40))
  b <- round(runif(8,1,80))
  bra1s <- c()
  bra1w <- c()
  bra2s <- c()
  bra2w <- c()

  for(i in 1:4){
    bra1s <- c(bra1s,summermm[a[i]])
    bra2s <- c(bra2s,summermm2[a[i]])
  }

  rm(i)

  for(i in 1:8){
    bra1w <- c(bra1w,wintermm[b[i]])
    bra2w <- c(bra2w,wintermm2[b[i]])
  }

  rm(i)
  skador1 <- c()
  l2=0
  while(l2<sum(bra1s) + sum(bra1w)){
    ru1 <- runif(1,0,1)
    if(ru1 < p){
      skador1 <- c(skador1,rlnorm(1,b1_sma_lnfit$estimate[1],b1_sma_lnfit$estimate[2]))
    }
    else{
      skador1 <- c(skador1,rnorm(1,mean(branch1_stora),sd(branch1_stora)))
    }
    l2 <- l2+1
  }

  sumskador <- c(sumskador,sum(skador1))
  z_xl <- c(z_xl,sum(sapply(skador1, function(x) max(x-k_xl,0))))
  meanz_xl <- c(meanz_xl,mean(z_xl))
```

```
  sum_min_yu1 <- c(sum_min_yu1,sum(sapply(skador1, FUN = function(x) min(x,k_xl))))

  z_sl <- c(z_sl,max(sum(skador1)-k_sl,0))
  meanz_sl <- c(meanz_sl,mean(z_sl))

  skador2 <- c()
  k=0
  while(k<sum(bra2s) + sum(bra2w)){
    ru <- runif(1,0,1)
    if(ru < p2){
      skador2 <- c(skador2,exp(rgamma(1,shape=b2_sma_lgamma$estimate[1],rate=b2_sma_lgamma$
    }
    else{
      skador2 <- c(skador2,rnorm(1,mean(branch2_stora),sd(branch2_stora))
    }
    k <- k+1
  }

  sumskador2 <- c(sumskador2,sum(skador2))
  z_xl2 <- c(z_xl2,sum(sapply(skador2, function(x) max(x-k_xl2,0))))
  meanz_xl2 <- c(meanz_xl2,mean(z_xl2))
  sum_min_yu2 <- c(sum_min_yu2,sum(sapply(skador2, FUN = function(x) min(x,k_xl2))))

  z_sl2 <- c(z_sl2,max(sum(skador2)-k_sl2,0))
  meanz_sl2 <- c(meanz_sl2,mean(z_sl2))
  l1 <- l1 +1
}

## slut simulering


## Distributions and quantile functions

## Distribution of total claim cost
par(mfrow=c(1,1))
tot_sum <- sumskador +sumskador2
hist(tot_sum,100)


## Distribution with XL cover
sum_mintot <- sum_min_yu1 +sum_min_yu2
```

```
hist(sum_mintot,50)

sum_xl <- sum_mintot + 1.1*(mean(z_xl) + mean(z_xl2))
hist(sum_xl,40)


## Distribution with sl cover
sum_minsl_1 <- sapply(sumskador, FUN=function(x) min(x,k_sl))
sum_minsl_2 <- sapply(sumskador2, FUN=function(x) min(x,k_sl2))

sum_min_totsl <- sum_minsl_1+sum_minsl_2
sum_sl <- sum_min_totsl + 1.1*(mean(z_sl) + mean(z_sl2))

hist(sum_sl,50)
hist(sum_min_totsl,50)
hist(tot_sum,50)


## z_sl3
k_sl3 <- k_sl+k_sl2
z_sl3 <- sapply(tot_sum, FUN = function(x) max(x-k_sl3,0))
mean(z_sl3)

sum_minsl3 <- sapply(tot_sum, FUN=function(x) min(x,k_sl3))
sum_sl3 <- sum_minsl3 + 1.1*mean(z_sl3)

hist(sum_sl3,40)
hist(sum_minsl3,50)
hist(tot_sum,100)

## XL

## DIst functions
sort_sumxl <- sort(sum_xl)
e_cdf1 <- 1:length(sort_sumxl) / length(sort_sumxl)

## Quantile function
se_q <- seq(from=0.9, to=0.999999,length.out = 1000)
qq_xl <- c()
for(i in se_q){
qq_xl <- c(qq_xl,sort_sumxl[which(e_cdf1 >= i)[1]])
```

```
}
plot(se_q,qq_xl)

## SL

sort_sumsl <- sort(sum_sl)
e_cdf2 <- 1:length(sort_sumsl) / length(sort_sumsl)
sort_sumsl[which(e_cdf2 >= 0.95)[1]]

## quantile function
qq_sl <- c()
for(i in se_q){
  qq_sl <- c(qq_sl,sort_sumsl[which(e_cdf2 >= i)[1]])
}
plot(se_q,qq_sl)

## Joint sl

sort_sumsl3 <- sort(sum_sl3)
e_cdf3 <- 1:length(sort_sumsl3) / length(sort_sumsl3)
sort_sumsl3[which(e_cdf3 >= 0.95)[1]]

## quantile function
qq_sl3 <- c()
for(i in se_q){
  qq_sl3 <- c(qq_sl3,sort_sumsl3[which(e_cdf3 >= i)[1]])
}
plot(se_q,qq_sl3)

par(mfrow=c(1,3))
plot(se_q,qq_xl,xlab = 'Probability', ylab = 'Quantiles', main= 'XL-cover')
plot(se_q,qq_sl,xlab = 'Probability', ylab = 'Quantiles', main= 'SL-cover')
plot(se_q,qq_sl3,xlab = 'Probability', ylab = 'Quantiles', main= 'Joint SL-cover')


par(mfrow=c(1,3))
plot(sort_sumxl,e_cdf1, xlab = 'Total claim cost', ylab = 'Probability', main = 'XL-cover')
plot(sort_sumsl,e_cdf2, xlab = 'Total claim cost', ylab = 'Probability', main = 'SL-cover')
plot(sort_sumsl3,e_cdf3, xlab = 'Total claim cost', ylab = 'Probability', main = 'Joint SL-
```