Collective Risk Model Description

Rudolfs Jansons SNR:2080485, Maxim de Vries SNR:2079287, Andrei Agapie SNR:2075694

2023-10-10

Contents

Introduction 3
Research Question 4
Data Description and Analyses 5
Models and Assumptions 6
Methods and Results 7
Possible Improvements 20
Conclusion 20

Introduction

In this report, we make use of collective insurance risk models to car insurance data. In the basis of insurance is managing uncertainty and assessing risk. Insurance is one of the fields where mathematical modeling meets real-world data. With the help of models like the one in this report insurance companies can offer financial protection to policyholders. As we go through the assignment , we will make use of collective insurance risk models to analyze historical car insurance data. These models offer insights into the aggregate claim amounts, probability distributions, and risk assessment measures All of these parameters are important for insurers decision making. To answer given questions we will use statistical techniques, simulation methods, and mathematical models to estimate parameters, assess risk levels, and explore strategies for risk mitigation. This report is made such that it provides a good overview of our findings. We will present our results systematically, making use of tables, figures, and easy to follow explanations. By the end of this report, we will have answered all of the research questions and evaluated the methods used. We will have offered possible improvements, supported by empirical evidence from the car insurance dataset. To conclude this introduction, our goal is to provide useful insights that can help to make strategic decision making easier.

Research Question

1. **Estimating Lambda**: How can this estimation be effectively accomplished using the techniques discussed in Section 2.3.3 of the course materials?
2. **Policy and Volume Analysis**: What is the total number of policies on the balance sheet at the beginning of 2023 (n)? What is the aggregate volume (v) ?
3. **Claim amount analysis**: How can we determine the values for 's' such that the probability of the aggregate claim amount (S) being less than or equal to 's' equals 0.9, 0.95, and 0.99, respectively? Using normal approximation and simulation approaches
4. **Capital at the start**: Given that the ruin probability without reinsurance equals 1%, what is the initial capital threshold (u) ?
5. **Proportional Reinsurance**: When proportional reinsurance is employed to reduce the default probability to 0.5%, what is the optimal proportionality factor (α) that minimizes risk while maximizing profit?
6. **Results and Interpretations**: How do the results obtained from different estimation approaches compare?
7. **Improvement Opportunities**: What additional information or methods could be used to improve the results pf the assignment?

Data Description and Analyses

In this assignment, we utilized a data set containing car insurance data spanning from the year 2000 to 2022.

Data set Name: Dataset_6

Source: The data set was made available on the Canvas page associated with this assignment.

Data Format: The data set is provided in a Comma-Separated Values (CSV) file format. It is structured into rows and columns, where each row represents an observation, and each column represents a variable or attribute.

Variables/columns:

1. **Year**: Calendar Year (integer number)
2. **Volume**: Volume of the policy in years, represented as a value in the interval [0, 1]
3. **NonzeroClaim**: Indicator for nonzero claims (1 = nonzero claim, 0 = no claim)
4. **ClaimSize**: Size of the claim. It is relevant when NonzeroClaim equals 1 (Amount > 0); otherwise, it is zero.
5. **Diesel**: Fuel Type (1 = Diesel, 0 = otherwise)
6. **Male**: Gender (1 = Male, 0 = non-male)
7. **Miles**: Annual number of miles driven (numeric value)

Data Integrity: For the purposes of this assignment, we assume that the dataset is accurate and complete.
Data Size: The dataset consists of 560,000 records and includes 7 columns.

Models and Assumptions

Collective Risk Model Description

In our analysis, we utilize the collective risk model to answer the research questions. This model characterizes the total claim amount, denoted as 'S,' over a one-year accounting period as follows:

$$S = Y_1 + Y_2 + \ldots + Y_N = \sum_{i=1}^{N} Y_i, \text{ where } S = 0 \text{ if } N = 0$$

1. **N (Number of Claims)**: 'N' is an integer random variable that counts the number of claims that occur over the one-year accounting period. 'N' can take on possible outcomes 0, 1, 2, and so on.
2. **Y1, Y2, . . . (Claim Sizes)**: 'Y1, Y2, . . . ' represent the individual claim sizes These claim sizes are independent and identically distributed (i.i.d.) random variables, denoted as 'G.' Importantly, 'G(0) = 0,' indicating that 'Yi' is always greater than 0 for all 'i.'
3. **Independence**: The random variables 'N' and 'Y1, Y2, . . . ' are independent of each other.

The collective risk model is basis to our analysis, as it allows us to characterize the total claim amount 'S'.

For a fixed volume $v > 0$ and a fixed expected claims frequency $\lambda > 0$, the random variable N follows a Poisson distribution, denoted as $N \sim \text{Poi}(\lambda v)$. The probability mass function p_k of the Poisson distribution is given by:

$$p_k = P[N = k] = \frac{e^{-\lambda v}(\lambda v)^k}{k!}, \quad k = 0, 1, 2, \ldots$$

The Poisson distribution $N \sim \text{Poi}(\lambda v)$ exhibits the following properties:

- **Expected Value (Mean)**: The expected value of N is given by $E[N] = \lambda v$.
- **Variance**: The variance of N is equal to the expected value, i.e., $\text{Var}(N) = \lambda v = E[N]$.

The total claim amount S follows a Compound Poisson Distribution, denoted as $S \sim \text{CompPoi}(\lambda v, G)$, if S has a compound distribution with the following components:

- $N \sim \text{Poi}(\lambda v)$: The number of claims ($N$) follows a Poisson distribution with parameters λ and v, where λ represents the expected claims frequency and v is the specified volume.
- Individual claim size distribution G: The distribution of individual claim sizes is represented by G. Each claim size follows the distribution G.

The Compound Poisson distribution $S \sim \text{CompPoi}(\lambda v, G)$ exhibits the following properties:

- **Expected Value (Mean)**: The expected value of S is given by $E[S] = \lambda v E[Y_1]$, where Y_1 represents the claim size of the first claim.
- **Variance**: The variance of S is calculated as $\text{Var}(S) = \lambda v E[Y_1^2]$, where Y_1^2 represents the square of the claim size of the first claim.

We need the following assumption when we solve the claim amount analysis with simulation:.

- Individual claim size distribution follows a gamma distribution. So $Y_i \sim \Gamma(\gamma, c)$

Methods and Results

To address the first research question, which involves estimating the parameter lambda (λ), we employed a method 2.3.3. Specifically, we calculated λ_{MV} using the formula:

To estimate lambda (λ) for the first research question, we used the method given in Section 2.3.3, where

$$\lambda_{\text{MV}} = \frac{\sum_{t=1}^{T} N_t}{\sum_{s=1}^{T} v_s}$$

we do this with the help of r

```
# Read the data from excel file
data <- read.csv("Dataset_6.csv", sep = ';')
# estimating lambda using technique in 2.3.3
lambda <- sum(data$NonzeroClaim) / sum(data$Volume)
```

```
## The estimated value of lambda is approximately 0.05901811
```

To answer the second question, we want to know the values of n and v.

```
# Filter the data for policies in 2022 with non-zero claims
filtered_data <- subset(data, Year == 2022 & NonzeroClaim == 1)

# Calculate the number of number of claims that are not zero
n <- nrow(filtered_data)

# Calculate the total volume (v) for 2022
total_volume_2022 <- sum(subset(data, Year == 2022)$Volume)

#Creating a subset of data where the year is 2022 and excluding claim amount so we can successfully fin
filtered_data2 <- subset(data, Year == 2022, select = -ClaimSize)
#Calculating n the amount of unique policy holder
n2 <- nrow(unique(filtered_data2))
```

```
## The number of non-zero claims in the year 2022  1067

## n, the number of policies on your balance sheet at the beginning of 2023  23323

## The total volume (v) for 2022 is  17134.5
```

From these calculations we can conclude that around 4,5% of police holders made a claim. To answer the third research question, we first needed to calculate the following:

- $E[y]$ (Mean of Claim Size)
- $E[y^2]$ (Second Moment of Claim Size)
- $E[S]$ (Expected Total Claim Amount)
- $Var(S)$ (Variance of Total Claim Amount)

We can not explicitly calculate the moments of the claim size since we do not it's distribution. So here we are approximating the moments by using the sample moments.

Calculation of Variables using models and assumptions

```
# Calculate the mean of claim size of filtered data (E[y])
mean_claim_size <- mean(filtered_data$ClaimSize)

# Calculate the second moment of claim size in the filtered data (E[y^2])
second_moment_claim_size <- sum(filtered_data$ClaimSize^2) / n
```

```r
# Calculate the expected total claim amount (E[S])
expected_total_claim_amount <- lambda * total_volume_2022 * mean_claim_size

# Calculate the variance of total claim amount (Var(S))
variance_total_claim_amount <- lambda * total_volume_2022 * second_moment_claim_size
```

we get the results

```
## Value of E[y]  22613.76
```

```
## Value of E[y^2]   625511249
```

```
## Value of E[S]  22868073
```

```
## Value of Var(S)  632545616231
```

Now we can apply normal approximation

$$P(S \leq s) = P\left(\frac{S - \ E(S)}{\sqrt{Var(S)}} \leq \frac{s - E(S)}{\sqrt{Var(S))}}\right) \approx \Phi\left(\frac{s - E(S)}{\sqrt{Var(S)}}\right)$$

$$s = \sqrt{\mathrm{Var}(S)} \cdot \Phi^{-1}(\alpha) + E(S)$$

For α belonging to the set $\{0.9, 0.95, 0.99\}$,

```r
# Calculate the 90th percentile total claim amount
s_09 <- sqrt(variance_total_claim_amount) * qnorm(0.9) + expected_total_claim_amount

# Calculate the 95th percentile total claim amount
s_095 <- sqrt(variance_total_claim_amount) * qnorm(0.95) + expected_total_claim_amount

# Calculate the 99th percentile total claim amount
s_099 <- sqrt(variance_total_claim_amount) * qnorm(0.99) + expected_total_claim_amount
```

Results of normal approximation. It is worth noting that its validity increases with volume as it is based on the Central Limit Theorem, converging in distribution as volume converges to infinity. We expected "s" to be larger than E[s] and this was confirmed in the result, with the small deviations that are anticipated appearing in our data as well. The simulation-approach yielded accurate results (also close to the normal approximation) for multiple reasons: due to the large number of constituent simulations, due to the underlying distribution being very close to Gamma and due to the thin-tailed nature of the Gamma distribution, which facilitates accuracy.

```
## Value of s for 0,9  23887326
```

```
## Value of s for 0,95  24176271
```

```
## Value of s for 0,99 24718282
```

Now we are looking at solving the third research question with simulations. First we are gonna calculate some intermediate steps with the R code down below.

We first load an important library that we are gonna need later. We also set a seed so that every time we reload the code, we do not get different results.

```r
#import handy library
library(MASS) #we use this to calc the MLE in gamma case

#setting an seed to ensure consistence
set.seed(1)
```

We are currently reformatting the data to utilize it in the code to address the question at hand.

```
#load in data
Dataset_6 <- read.csv("Dataset_6.csv", sep=";")
#get all non zero claims
Dataset_no_zero_claim <- subset(data, NonzeroClaim ==1)
#get non zero claims in year 2022
Dataset_2022_no_zero_claim <- subset(Dataset_no_zero_claim, Year ==2022)
#store the claim size of 2022
data_claim_size <- Dataset_2022_no_zero_claim$ClaimSize
```

We assumed that the claim size follows a gamma distribution, but we lack knowledge of its parameters. Consequently, we employ estimation techniques, specifically the Maximum Likelihood and Method of Moments estimators, utilizing non-zero claim sizes from 2022 for this purpose.

First, we calculate the ML estimates using the fitdistr function from the MASS library. This function employs a numerical approach for estimation. However, it may not perform optimally when dealing with data comprising large values. To address this issue, we rescale both the data and then scale the estimates back.

```
#let's calculate the ML estimates
est <- fitdistr(data_claim_size/1000, "gamma") #we scale the data
gamma_est <- est$estimate #save the estimates
gamma_est['rate'] <- gamma_est['rate'] /1000 #rescale the rate parameter
```

We are now calculating the MM estimates using the formula that was derived during class.

The formula that we use is:

$$\hat{c}^{MM} = \frac{\hat{\mu}_n}{\hat{\sigma}_n^2} \text{ and } \hat{\gamma}^{MM} = \frac{\hat{\mu}_n^2}{\hat{\sigma}_n^2}$$

```
#let's calculate the MM estimates
gamma_est_mm <- est$estimate #save the estimates in this way to keep the named vect
gamma_est_mm['rate'] <- mean(data_claim_size)/var(data_claim_size)
gamma_est_mm['shape'] <- mean(data_claim_size)^2/var(data_claim_size)
```

This table shows the ML and MM estimates that we found.

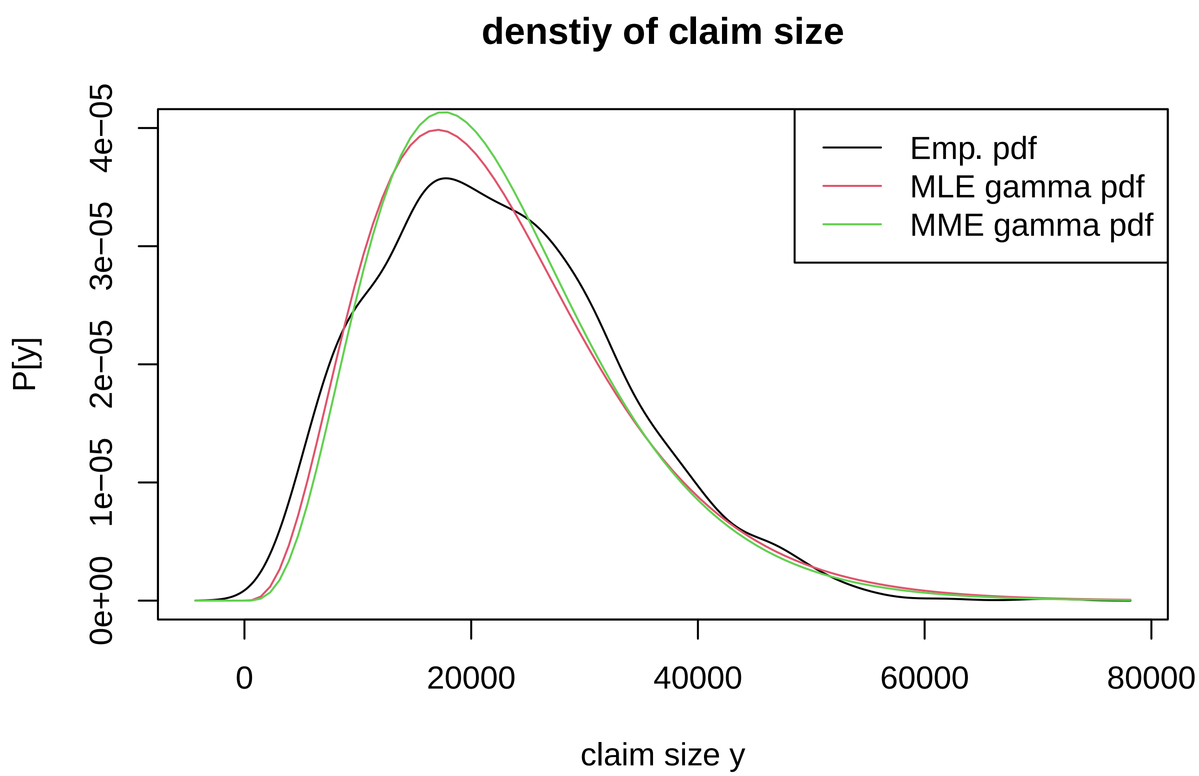
	ML est.	MM est.
shape	4.0605173	4.476543
rate	0.0001796	0.000198

From the table we can see that the ML and MM estimates are not completely different. We would say that they are similar.

We aim to assess the validity of assuming that the claim sizes follow a gamma distribution. While we won't provide a formal proof, we intend to visualize the distribution. Specifically, we will create plots representing the claim size distribution under the gamma assumption and compare it with maximum likelihood (ML) and method of moments (MM) estimations. Additionally, we will utilize data from 2022 to construct an Empirical Probability Density Function, employing R's density function for this purpose. We will then use the Empirical pdf as a benchmark to assess how closely the results align with our estimated gamma distribution. This is of course not a formal way of testing but it will give use some insight.

```
#here we plot the Empirical Probability Density Function with density
den <- density(data_claim_size)
plot(den$x, den$y, type='l',ylim = c(0,4/100000),xlab = 'claim size y',ylab = 'P[y]'
     ,main = 'denstiy of claim size')
```

```r
#here we draw the estimated gamma distribution of the claim size
curve(dgamma(x, gamma_est['shape'], gamma_est['rate']), add=TRUE, col=2) #mle case
curve(dgamma(x, gamma_est_mm['shape'], gamma_est_mm['rate']), add=TRUE, col=3) #mme case

#make a legend
legend("topright",legend = c("Emp. pdf","MLE gamma pdf",'MME gamma pdf'),col=c(1,2,3)
       ,lty=c(1,1,1))
```



In this plot, we observe that the estimated PDFs exhibit a similar shape to the Emp. PDF, which lends validity to the assumption that the claim size follows a gamma distribution. Additionally, the PDFs obtained through MLE and MME are closely aligned, as evident from the estimated parameters.

Now we will start with the Simulation. We are first going to prepare some variable and functions.

```r
#prepare some things for the simulation
n_simulations <- 100000 #In the project description, this was called J
poi_par_est <- lambda * total_volume_2022
S_mle <- rep(NaN,n_simulations)#vect will contain simulated total claim amounts in MLe
S_mme <- rep(NaN,n_simulations)#vect will contain simulated total claim amounts in MMe
```

Now, we create a function that calculates the total claim amount for each scenario. This function will received a random number drawn from the Poisson distribution, which determines how many times we will draw from our estimated gamma distribution. We then have all the simulated claim sizes, which we sum up and return as the aggregate claim amount for 2023 in this one simulation.

```r
#we give this function a number that indicate how many claims we had in scenario j
#parameter wich is 1 if we use mle and 0 if we use mme
Func_Monte_Sim_Claim_Size = function(n,wich)
```

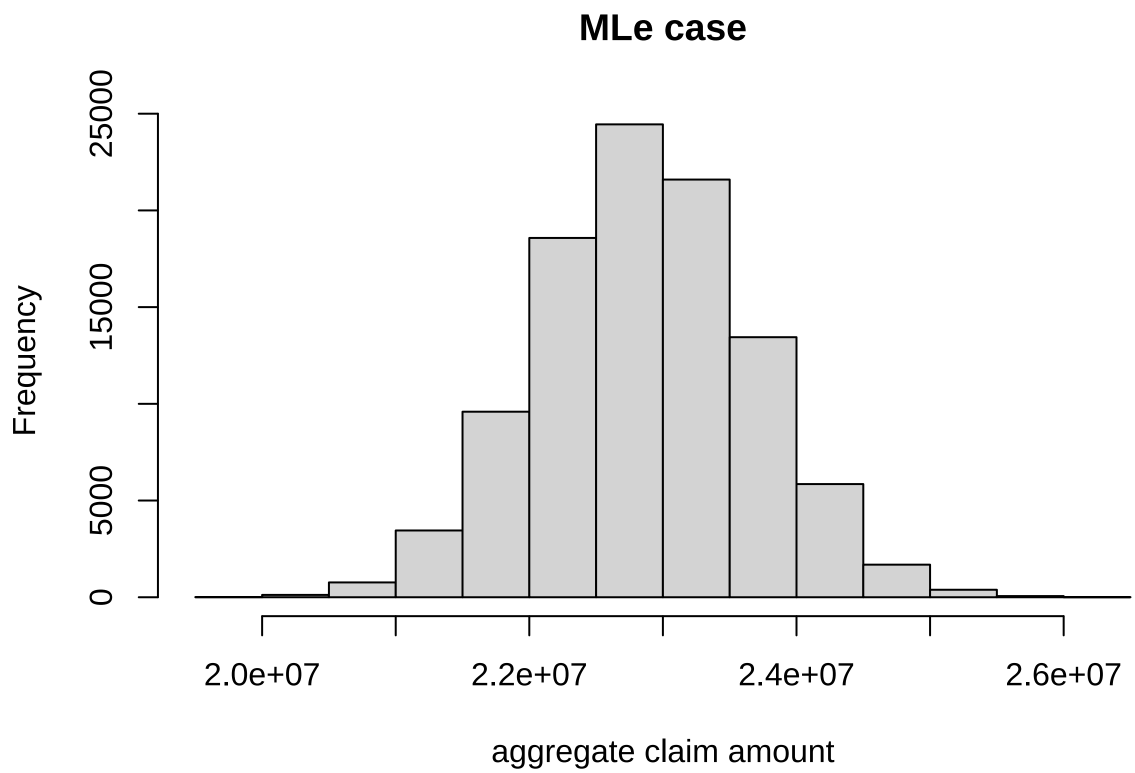
```
{
  #Draw n observations from our gamma distribution of the claim size estimated
  if (wich ==1){
    y_j <- rgamma(n, gamma_est['shape'], gamma_est['rate'])
  }
  else {
    y_j <- rgamma(n, gamma_est_mm['shape'], gamma_est_mm['rate'])
  }
  #calculating the total claim amount in scenario j
  S_j <- sum(y_j)
  return(S_j)
}
```

We will now conduct a Monte Carlo Simulation involving 100,000 simulations. Initially, we draw a number from a Poisson distribution using the earlier estimated parameter, which represents the number of claims in each simulation. Subsequently, we utilize a custom function to calculate the aggregate claim amount.

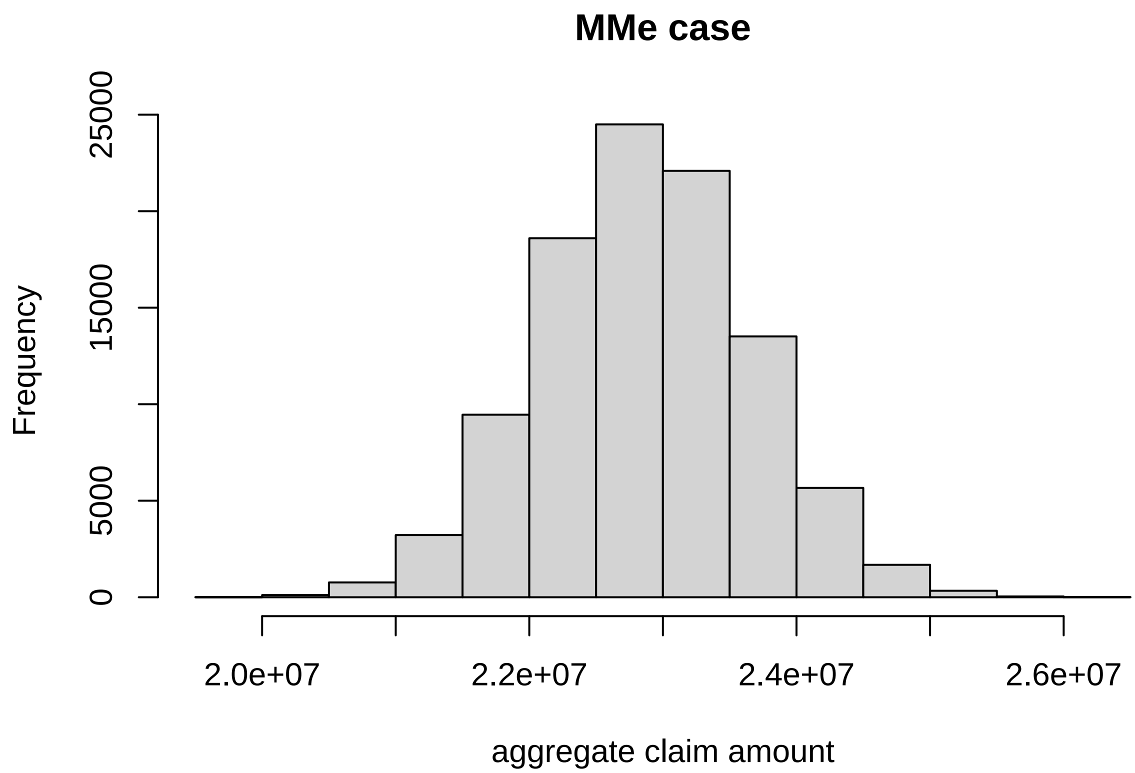
```
#now draw a bunch of times from poisson dist. with the earlier estimated parameter.
#this contains all number of non-zero claims in all scenario's
N_claims_per_scenario <- rpois(n_simulations,poi_par_est)

#we going to loop through all scenarios and get total claim amount in that simulation
for (i in 1:n_simulations) {
  #store all total claim amounts in mle case
  S_mle[i] <- Func_Monte_Sim_Claim_Size(N_claims_per_scenario[i],1)
  #store all total claim amounts in mme case
  S_mme[i] <- Func_Monte_Sim_Claim_Size(N_claims_per_scenario[i],0)
}
```

We make some histograms of the simulated aggregate claim amount.

```
#histogram to view how the S look like in mle case
hist(S_mle,xlab ='aggregate claim amount',main = 'MLe case')
```




```r
#histogram to view how the S look like in mme case
hist(S_mme,xlab ='aggregate claim amount',main = 'MMe case')
```



We will now create an Empirical Cumulative Distribution Function for the aggregate claim amount distribution in 2023.

We first make a function that can calculate the Emp. CDF.

```
Func_E_CDF <- function(dataset) {
  #get useful info
  n_obs <- length(dataset)
  data_sorted <- sort(dataset)
  e_cdf <- rep(0, n_obs) #make vect to store results in

  #loop trough all data and calc the e CDF with the formula
  for (i in 1:n_obs) {
    e_cdf[i] <- sum(data_sorted <= dataset[i]) / n_obs
  }
  #this should now be the values of the E CDF
  return(e_cdf)
}
```

Now we use this function to calculate the CDF value's for the simulated claim amount

```
S_e_CDF_mle <- Func_E_CDF(S_mle)
S_e_CDF_mme <- Func_E_CDF(S_mme)
```

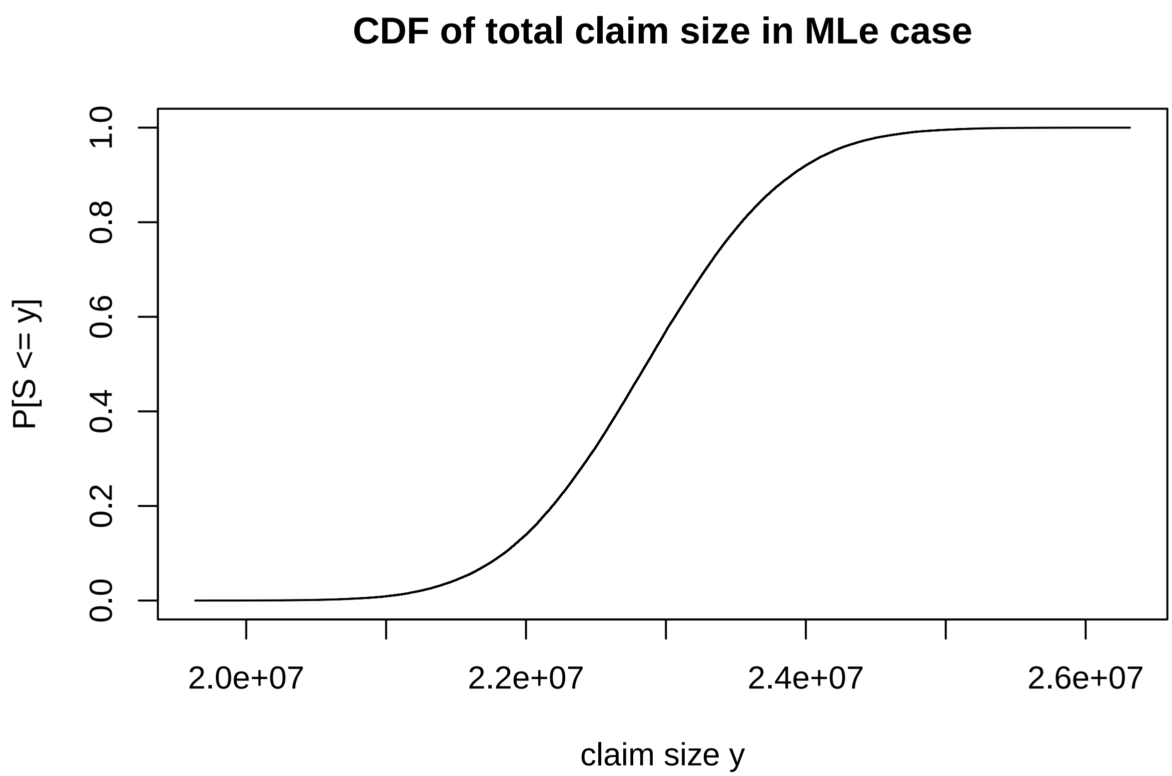
Now with this CDF we can estimate the values for s such that P[S $\leq$ s] equals 0.9, 0.95 and 0.99.

First we are going to sort our data

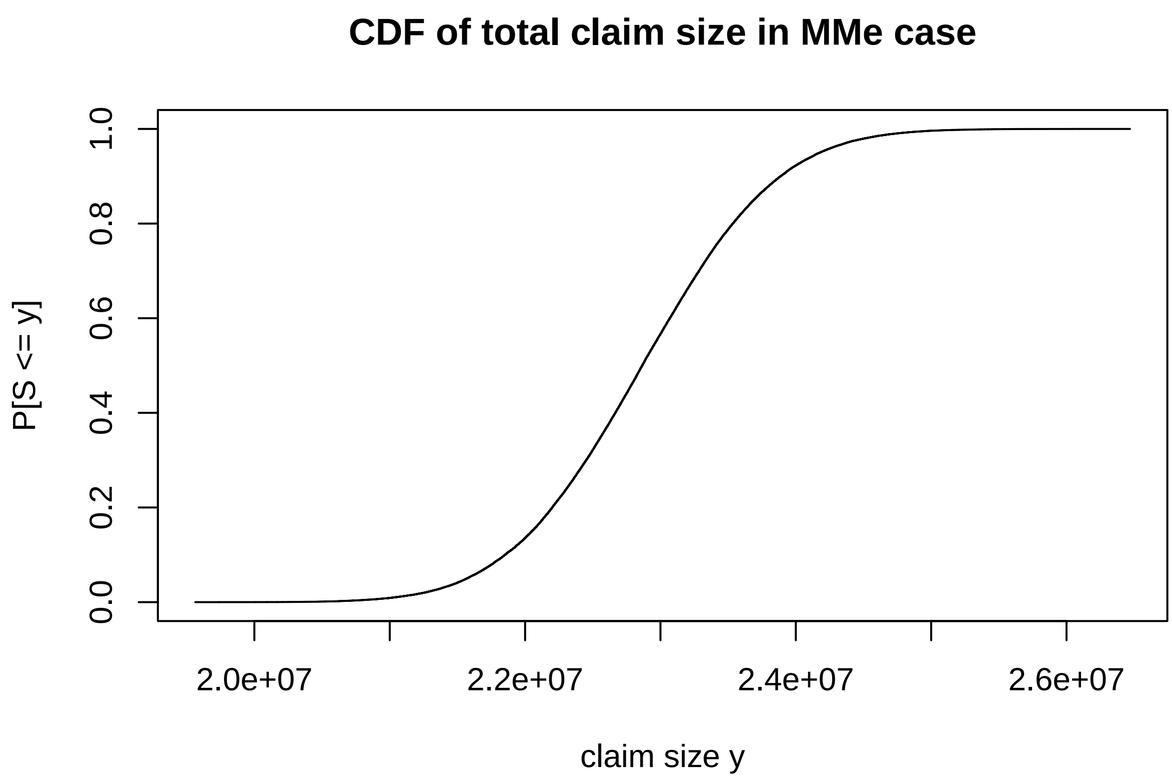
```
sort_S_mle <- sort(S_mle)
sort_S_e_CDF_mle <- sort(S_e_CDF_mle)

sort_S_mme <- sort(S_mme)
sort_S_e_CDF_mme <- sort(S_e_CDF_mme)
```

Now we can plot how the Emp. CDF looks like in our simulation.

```
plot(sort_S_mle,sort_S_e_CDF_mle, type = "s",xlab = 'claim size y',ylab = 'P[S <= y]'
     ,main = 'CDF of total claim size in MLe case')
```




```
plot(sort_S_mme,sort_S_e_CDF_mme, type = "s",xlab = 'claim size y',ylab = 'P[S <= y]'
     ,main = 'CDF of total claim size in MMe case')
```



Now we use the which function to find the s that we are searching for.

```
#Now we search for the first S such that P(S) >= 0.9(or 0.95/0.99)
#first with mle with this code
s_0.9_mle <- sort_S_mle[which(sort_S_e_CDF_mle >= 0.9)[1]] #where s = 0.9
s_0.95_mle <- sort_S_mle[which(sort_S_e_CDF_mle >= 0.95)[1]] #where s = 0.95
s_0.99_mle <- sort_S_mle[which(sort_S_e_CDF_mle >= 0.99)[1]] #where s = 0.99

#then with mme
s_0.9_mme <- sort_S_mme[which(sort_S_e_CDF_mme >= 0.9)[1]] #where s = 0.9
s_0.95_mme <- sort_S_mme[which(sort_S_e_CDF_mme >= 0.95)[1]] #where s = 0.95
s_0.99_mme <- sort_S_mme[which(sort_S_e_CDF_mme >= 0.99)[1]] #where s = 0.99
```

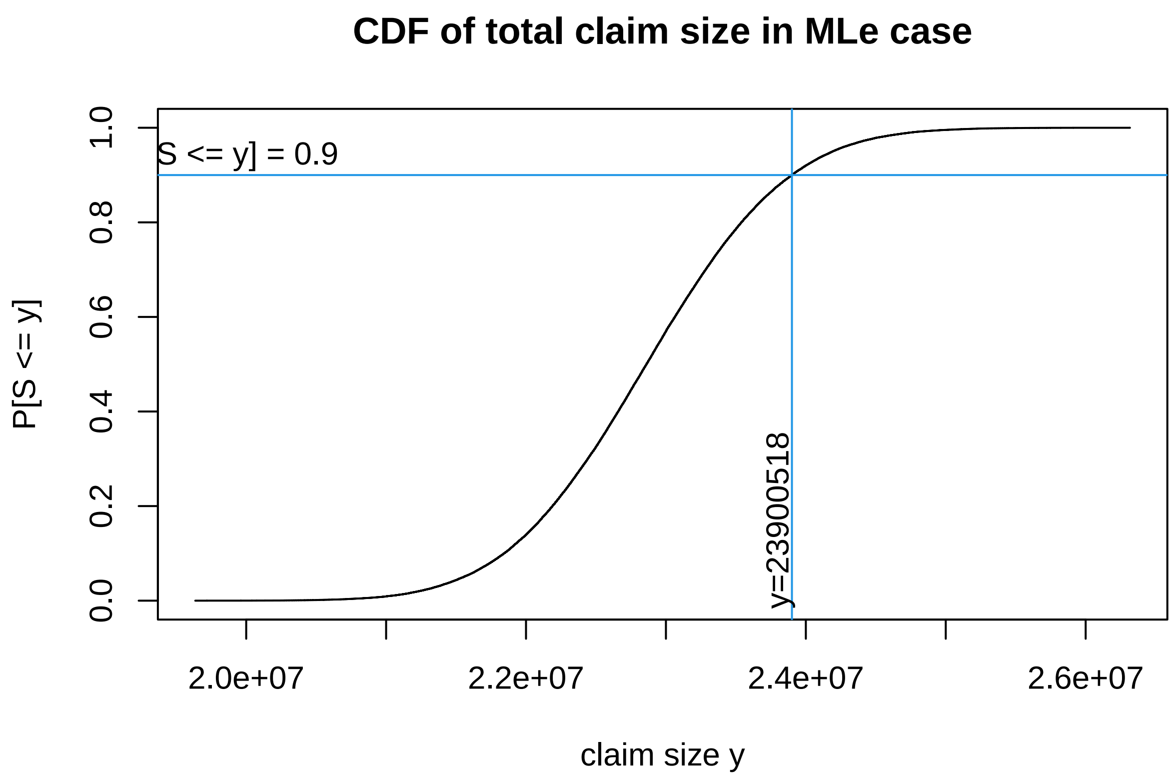
Results of simulation.

```
library(knitr)
kable(cbind(c(0.9,0.95,0.99),c(s_0.9_mle,s_0.95_mle,s_0.99_mle),c(s_0.9_mme,s_0.95_mme,s_0.99_mme)),col
```

level	s in MLe	s in MMe
0.90	23900518	23887308
0.95	24192095	24177199
0.99	24751178	24729817

let's plot one of these point to see if the answer seems plausible

```
plot(sort_S_mle,sort_S_e_CDF_mle, type = "s",xlab = 'claim size y',ylab = 'P[S <= y]'
     ,main = 'CDF of total claim size in MLe case')
abline(h = 0.9, col = 4)
text(x=19900000, y=0.94, 'P[S <= y] = 0.9')
abline(v = s_0.9_mle, col = 4)
text(x=23820002, y=0.17, srt=90, 'y=23900518')
```



Now we have solved Claim amount analysis

The results are:

```
kable(cbind(c(0.9,0.95,0.99),c(s_09,s_095,s_099),c(s_0.9_mle,s_0.95_mle,s_0.99_mle),c(s_0.9_mme,s_0.95_
```

level	s in norm. aprox.	s in sim. MLe	s in sim. MMe
0.90	23887326	23900518	23887308
0.95	24176271	24192095	24177199
0.99	24718282	24751178	24729817

Given the similarity between the MLE and the MME CDF's, we can infer for our assumption over the claim distribution has a modicum of additional justification.

Now we will solve the research question about proportional reinsurance

We are first going to find our initial capital.

We load an important library that we are gonna need later.

```r
#import handy library
library(latticeExtra) #we use this to make the plots later
```

```
## Loading required package: lattice
```

First we are gonna calculate some intermediate steps, we start with the total incoming premium.

$$\pi = (1+\theta)E[S]$$

where θ is the safety loading.

```r
#Set the safety loading
safety_loading <- 0.05
#Now we calc the total incoming premium
total_inc_premium <- (1+safety_loading)*expected_total_claim_amount
```

Now we can calculate our initial capital by using an normal approximation.

$$P(ruin) = P(S > \pi{+}u) = 1{-}P(S \leq \pi{+}u) = 1{-}P\left(\frac{S - \ E(S)}{\sqrt{Var(S)}} \leq \frac{\pi + u - E(S)}{\sqrt{Var(S))}}\right) \approx 1{-}\Phi\left(\frac{\pi + u - E(S)}{\sqrt{Var(S)}}\right)$$

so in our case we solve

$$\frac{\pi + u - E(S)}{\sqrt{Var(S)}} = \Phi^{-1}(0.99)$$

to get our formula for initial capital u

$$u = \Phi^{-1}(0.99) * \sqrt{Var(S)} + E(S) - \pi$$

```r
# Calculate the initial capital (u) by using normal approximation
ini_cap_1 <- sqrt(variance_total_claim_amount) * qnorm(0.99) #split in 2 part
ini_cap_2 <- expected_total_claim_amount - total_inc_premium #to make fil
initial_capital <- ini_cap_1 + ini_cap_2
```

So our initial capital u is `rinitial_capital`‘

Now we can start with our calculation when we use reinsurance. We first need to set some values in the r code. We are making a vector that contains a lot of possible proportionality factor α

```r
#Now we start reinsuring
pos_alpha <- seq(0, 1, by=0.001) #setting all alpha that we are evaluating
n_of_pos_alpha <- NROW(pos_alpha)
saf_load_reinsurer <- 0.07
# vect for storing all expected profit per alpha
total_expected_profit_alpha <- rep(NaN,n_of_pos_alpha)
# vect for storing all total premium per alpha
t_inc_prem_al <- rep(NaN,n_of_pos_alpha)
# vect for storing all total premium per alpha
ruin_prob_alpha <- rep(NaN,n_of_pos_alpha)
```

For each α we will look at, the expected and variance of total claim amount for the cedent at that level of reinsurance. We will use this to get the expected profit and the ruin probability that we have when we reinsurance at this level.

To calculate the ruin probability with reinsurance level α we use the following formula.

$$P(ruin) = P(S^c > \pi^c{+}u) = 1{-}P(S^c \leq \pi^c{+}u) = 1{-}P\left(\frac{S^c - \ E(S^c)}{\sqrt{Var(S^c)}} \leq \frac{\pi^c + u - E(S^c)}{\sqrt{Var(S^c))}}\right) \approx 1{-}\Phi\left(\frac{\pi^c + u - E(S^c)}{\sqrt{Var(S^c)}}\right)$$

where

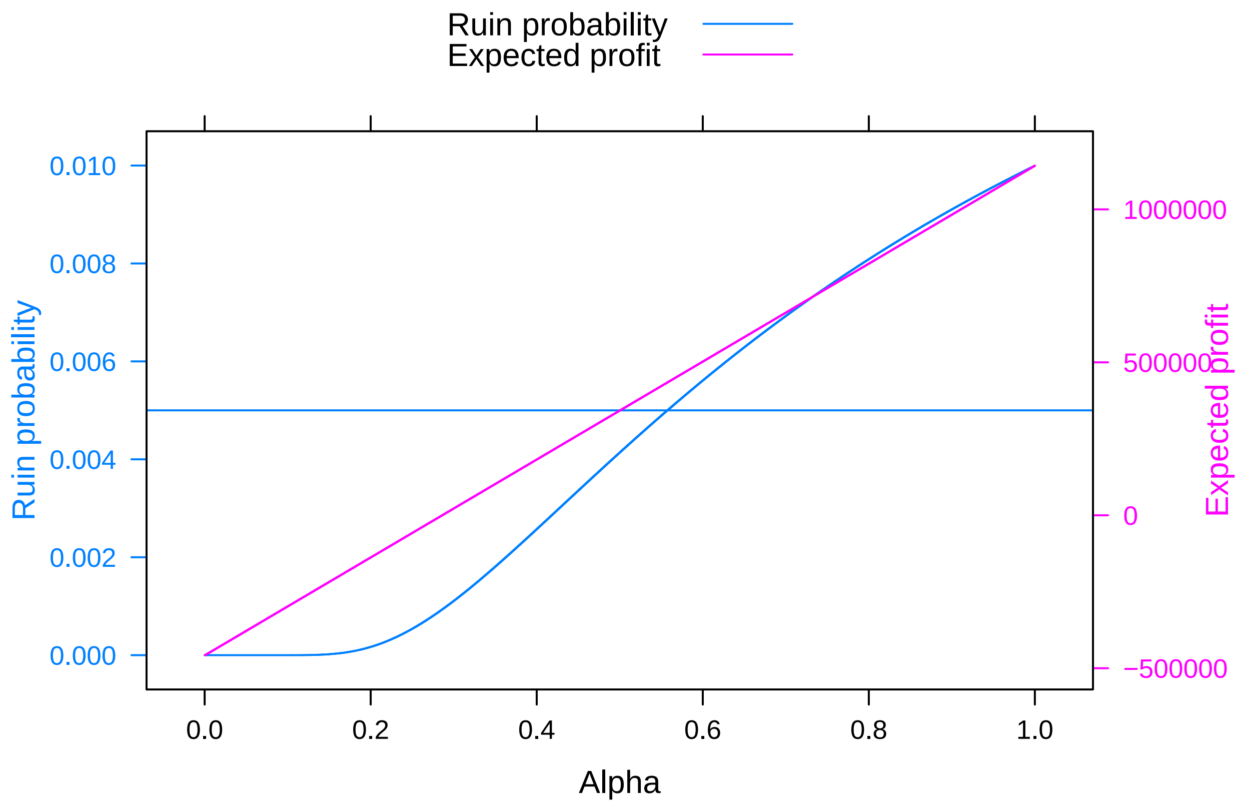
$$E(S^c) = \alpha * E(S) \text{ , } Var(S^c) = \alpha^2 * Var(S) \text{ and } \pi^c = [1 + \theta - (1 + \xi) * (1 - \alpha)] * E(S)$$

The code for the implementation is:

```
#now we loop trough all possibly alpha's and then calculate the ruin probability
for (i in 1:n_of_pos_alpha) {
  alpha <- pos_alpha[i]
  #calc the expected total claim amount of the cedent(us) (E[S^c])
  expected_total_claim_amount_alpha <- expected_total_claim_amount * alpha
  # Calculate the variance of total claim amount of the cedent (Var[S^c])
  variance_total_claim_amount_alpha <- variance_total_claim_amount * alpha**2

  #Now we calc the total incoming premium of the cedent
  t_inc_prem_al[i]<-(1+safety_loading-(1+saf_load_reinsurer)*(1-alpha))*expected_total_claim_amount

  #now we calc the expected profit
  total_expected_profit_alpha[i] <- t_inc_prem_al[i] - expected_total_claim_amount_alpha

  #now we can calculate the ruin probability when we have an alpha level of reinsurance
  sd_var_S <-  sqrt(variance_total_claim_amount_alpha)
  ruin_p_1 <- (initial_capital + t_inc_prem_al[i] - expected_total_claim_amount_alpha) / sd_var_S
  ruin_prob_alpha[i] <- 1 - pnorm(ruin_p_1)
}
```

We will now show a plot of the ruin probability and expected profit over all the α that we evaluate.

```
data <- data.frame(pos_alpha,ruin_prob_alpha,total_expected_profit_alpha)

#make separate plots for each part of 2 y scale graph
graph1 <- xyplot(ruin_prob_alpha ~ pos_alpha, data, type = "s"
                 , lwd=1,abline=c(h=0.005),xlab = 'Alpha',ylab = 'Ruin probability')
graph2 <- xyplot(total_expected_profit_alpha ~ pos_alpha, data, type = "s"
                 , lwd=1,xlab = 'Alpha',ylab = 'Expected profit')

# --> Make the plot with second y axis:
doubleYScale(graph1, graph2, text = c("Ruin probability", "Expected profit")
             , add.ylab2 = TRUE)
```



Now we can find the maximal profit for witch the ruin probability is 0.05. The optimal α can be found by search for the α that has an ruin probability of 0.05. The following code can find it

```
opt_alpha <- pos_alpha[which(ruin_prob_alpha >= 0.005)[1]]
opt_profit <- total_expected_profit_alpha[which(ruin_prob_alpha >= 0.005)[1]]
```

This is an approximation of the true alpha since this code is not looking trough all possible opt_profit however the answer should be good enough.

The optimal α is: 0.557 this has an profit of: 4.3426471×10^5

Possible Improvements

Within the simulation, assuming that the claim size follows a gamma distribution is a strong assumption, we informally examined the validity of this assumption. It necessary to subject it to a more formal test. Additionally, it might be beneficial to explore if other distributions for claim sizes are more reasonable. We could also consider a non-parametric approach to estimate the claim size distribution and use a simulation-based approach in order to determine the optimal alpha. We are currently estimating E(Y) by it's sample analog and then use this to find E(S). In general we could modify our approach by assuming a certain distribution for the individual claim sizes, and then use this to estimate E[S]/E[Y].

Conclusion

To conclude, we have answered all the questions of the assignment:we have estimated lambda by method of moments,stated the number of policies and volume, determined the demanded value of "s" so that probability that the aggregate claim amount being less than equal to it would be 0.9, 0.95 and 0.99 respectively and provided an initial capital threshold u and an optimal proportionality factor alpha.