A Simulation of Individual Insurance Claims

12 December 2019

Question 1

Question 1, Part (a)

Consider a random variable X that follows a Pareto distribution. The density of X is given by:

$f(x) = \frac{\alpha\beta^\alpha}{(x+\beta)^{\alpha+1}}$ for $x \geq 0$,

$f(x) = 0$ for $x < 0$,

where the constants $\alpha > 0$, $\beta > 0$. These requirements on the constants were not given in the exercise but follow easily. We have $\int_0^\infty f(x)\,dx = 1$ if and only if $\alpha > 0$ and consequently $f(x) \geq 0$ for all x if and only if $\beta > 0$.

Step 1. Calculate the cumulative distribution function $F(x)$ of X. For $x \geq 0$, we have:

$$F(x) = P(X \leq x) = \int_0^x f(y)\,dy = \int_0^x \alpha\beta^\alpha (y+\beta)^{-\alpha-1}\,dy = [-\beta^\alpha (y+\beta)^{-\alpha}]_0^x = 1 - \frac{\beta^\alpha}{(x+\beta)^\alpha}$$

Therefore, the cumulative distribution function $F(x)$ of X is given by:

$F(x) = 1 - \frac{\beta^\alpha}{(x+\beta)^\alpha}$ for $x \geq 0$,

$F(x) = 0$ for $x < 0$.

Step 2. Calculate the expectation $E[X]$.

$$E[X+\beta] = \int_0^\infty (x+\beta)\frac{\alpha\beta^\alpha}{(x+\beta)^{\alpha+1}}\,dx = \int_0^\infty \alpha\beta^\alpha (x+\beta)^{-\alpha}\,dx = \left[\frac{\alpha\beta^\alpha}{1-\alpha}(x+\beta)^{1-\alpha}\right]_0^\infty = \frac{\alpha\beta}{\alpha-1}$$

Where we imposed the condition $\alpha > 1$ so that $\lim_{x\to\infty}(x+\beta)^{1-\alpha} = 0$.

Consequently, $E[X] = E[X+\beta-\beta] = E[X+\beta] - \beta = \frac{\alpha\beta}{\alpha-1} - \beta = \frac{\beta}{\alpha-1}$

It is obvious that if $\alpha < 1$, the mean will be infinite. If $a = 1$, the primitive will include the term $log(x+\beta)$. Since $\lim_{x\to\infty} log(x+\beta) = \infty$, the mean would be infinite in this case as well.

Step 3. Calculate the median of X.

The median of X, denoted by x_m, solves the equation $F(x_m) = 0.5$. We get:

$$1 - \frac{\beta^\alpha}{(x_m+\beta)^\alpha} = 0.5$$

$$\Rightarrow (x_m+\beta)^\alpha = 2\beta^\alpha$$

$$\Rightarrow x_m = (2^{\frac{1}{\alpha}} - 1)\beta$$

Step 4. Calculate the variance $var(X)$.

To determine the variance, we start by calculating $E[(X+\beta)^2]$.

$$E[(X+\beta)^2] = \int_0^\infty (x+\beta)^2 \frac{\alpha\beta^\alpha}{(x+\beta)^{\alpha+1}} dx = \int_0^\infty \alpha\beta^\alpha (x+\beta)^{1-\alpha} dx = \left[\frac{\alpha\beta^\alpha}{2-\alpha}(x+\beta)^{2-\alpha}\right]_0^\infty$$

When $\alpha > 2$, we get $E[(X+\beta)^2] = \frac{\alpha\beta^2}{\alpha-2}$. Then,

$$E[X^2] = E[(X+\beta)^2] - 2\beta E[X] - \beta^2 = \frac{\alpha\beta^2}{\alpha-2} - \frac{2\beta^2}{\alpha-1} - \beta^2 = \frac{2\beta^2}{(\alpha-2)(\alpha-1)}$$

Using the results from above, we can calculate $var(X)$ as follows:

$$var(X) = E[X^2] - E[X]^2 = \frac{2\beta^2}{(\alpha-2)(\alpha-1)} - \frac{\beta^2}{(\alpha-1)^2} = \frac{\alpha\beta^2}{(\alpha-1)^2(\alpha-2)}$$

Question 1, Part (b)

Consider $U \sim Unif(0,1)$. Let F_x be the cumulative distribution function of X, which is assumed to be strictly increasing. Therefore, the function F_x is invertible.

The cumulative distribution function of the random variable ${F_x}^{-1}(U)$is then given by the following:

$$F_{F^{-1}(U)}(x) = P\left[{F_X}^{-1}(U) \leq x\right] = P[U \leq F_X(x)] = F_x(x)$$

So, the random variable ${F_x}^{-1}(U)$ has the same distribution as X. Consequently, to simulate from X we generate $U \sim Unif(0,1)$ and compute $X = {F_X}^{-1}(U)$.

Specifically, for the Pareto CDF F found above, the inverse $F^{-1}: (0,1) \to \mathbb{R}$ is given by:

$$F^{-1}(u) = \frac{\beta}{(1-u)^{\frac{1}{\alpha}}} - \beta$$

Question 1, Part (c)

A brief summary of our methodology is as follows:

Step 1. We use the inversion method to simulate from X. Our first step is to create a function *invpar* that calculates the inverse Pareto CDF $F^{-1}(u)$ as defined above.

Step 2. We create a function *rpar* which returns n simulated values from a Pareto distribution with parameters α, β. The first part of the function generates n random values from a uniform distribution on (0,1), and the second part runs these values through the *invpar* function described in step 1.

Below is a histogram of our simulated values:

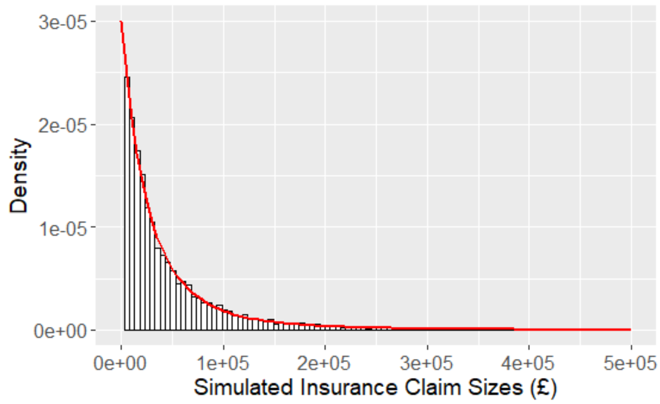

Figure 1.1. *Histogram of 10,000 values drawn from X. As you can see, the distribution is heavy-tailed. Note: our simulated values range from 3.5 to about 2.3 million. For aesthetic purposes, we imposed an x-axis limit of 500,000 on the histogram, which means that 38 of our observations were excluded. However, the histogram still includes over 99.5% of our simulated values. Therefore, it provides an accurate (and significantly less compressed) picture of the distribution of our simulated values drawn from X.*

Question 1, Part (d)

Note that insurance claims are nonnegative and unbounded, so a support of $[0,\infty)$ is appropriate. The most important property of the Pareto distribution is that it exhibits fat tails. In insurance, one must account for the possibility of large claims. Most realizations from a Pareto distribution are relatively small, but there are still a significant number of large claims. For example, an exponential distribution is not suitable (the probability of large claims is too small).

Question 2

In the function *assets1*, we simulate the assets at year-end by doing the following:

Step 1. Simulate for each client a value 0 (no claim) or 1 (claim). After all, we assume that each client makes at most 1 claim per year. The probability of having 1 claim is given to be 0.1. To simulate one value from this discrete distribution, we simulate a value from a uniform distribution on (0,1). The {0,1} value can be simulated by plugging the result into the following:

$g(u) = 1$ if $u \leq 0.1$,

$g(u) = 0$ if $u > 0.1$.

In this step, we assume that the event of making a claim or not is independent across the clients. We sum the binary values indicating whether a client made claim to get the total number of claims (*#claims*).

Step 2. The next step is to simulate the total value of the claims in a year. Using the *rpar* function from question 1, we simulate *#claims* values from a Pareto distribution with α = 3 and β = 100,000. In this step, we assume that the value of the claims is independent. Summing the values of these claims in a year gives the total value of claims in a year.

Step 3. The assets at year-end can then be computed as:

$assets_1 = assets_0 + NC * premium - total\ claims,$ where *NC* = number of clients.

Note that the *sapply* function is included to deal with an input vector of *probclaim*. The function *assets1* will then return a vector with the assets at year-end for each value of *probclaim*. Also, when the input premium is a vector, *assets1* returns a vector. We use the function *MCSimul* to generate many assets at year end. This function repeatedly calls *assets1* and returns all results. Using the *summary* function, we get the following:

Table 2.1. Distribution of Simulated Assets at Year End in £ Millions

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
(£14.00)	£0.69	£1.34	£1.24	£1.91	£4.10

Note that the expected assets at year-end is the mean value of £1.24 million.

Step 4. Lastly, the insurance firm is bankrupt when assets at year-end are negative. Here, we make use of the information that premiums are paid at the start of the year. Assets are therefore nonincreasing throughout the year, so bankruptcy can be assessed by considering the assets at year end. By counting the number of simulations of assets at year-end that are negative and dividing by the total number of simulations, we estimate the probability of bankruptcy at 9.8%.

In addition, we carry out some supplementary analysis to expand the current scope of our work. The details of these extra simulations are explained in Appendix B.

Question 3

Question 3, Part (a)

We start by generating a sequence of premiums £5500, £5750, £6000, …, £8000. Next, we use the *MCSimul* function to generate many assets at year-end. Lastly, we calculate the probability of bankruptcy at year end for each annual premium in accordance with the methodology outlined in question 2.

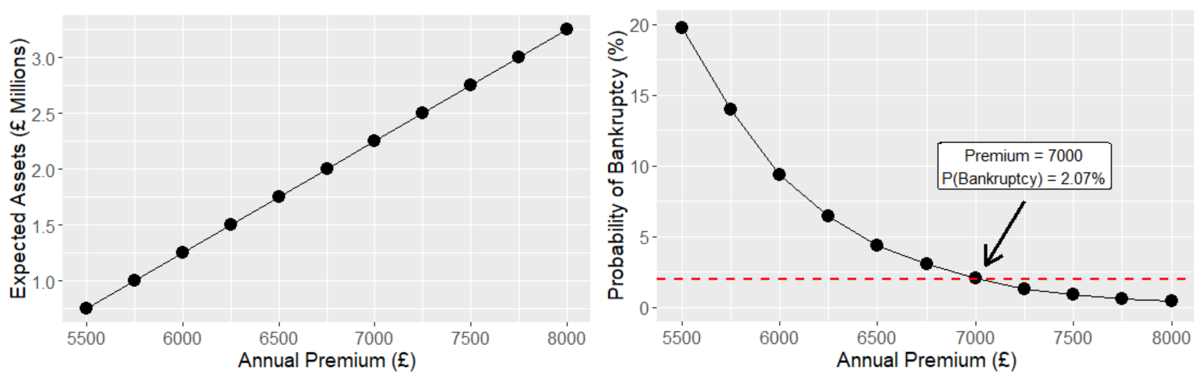

Figure 3.1. *Left: expected assets (£ millions) at year end for varying annual premiums. As anticipated, the expected assets increases as annual premiums increase. Right: probability of bankruptcy at year end (%) vs. annual premium (£). If the risk of bankruptcy (all else equal) should be no more than about 2%, the annual premium should not exceed £7,000.*

Question 3, Part (b)

We start by generating a sequence of the probabilities of any customer making a claim 5.0%, 5.5%, 6.0%, …, 15.0%. Next, we use the *MCSimul* function to generate many assets at year-end. Lastly, we calculate the probability of bankruptcy at year end for each probability in accordance with the methodology outlined in question 2.

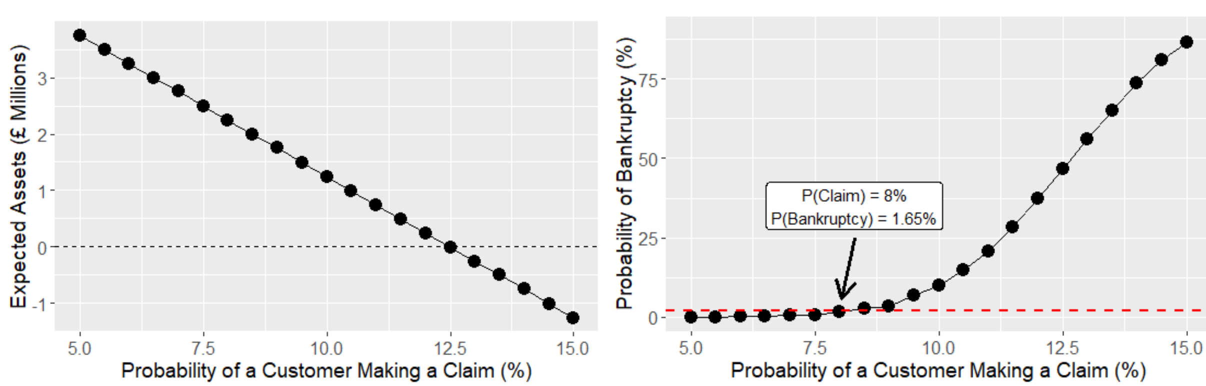

Figure 3.2. *Left: expected assets (£ millions) at year-end for varying probabilities of a customer making a claim. Obviously, expected assets decrease when the probability of a customer making a claim is high. Right: probability of bankruptcy at year-end (%) vs. probability of a customer making a claim (%). If the risk of bankruptcy (all else equal) should be no more than about 2%, the probability of a customer making a claim would need to be lower than about 8%. If the probability of a customer making a claim jumps to 8.5%, then the risk of bankruptcy increases almost 1%-point to 2.60%. Simulations considering the probability of making a claim between 8% and 8.5% with a smaller step-size suggest the probability of a customer making a claim would need to be lower than about 8.2% for the probability of bankruptcy not be no more than 2% (but keep in mind, these values are based on simulations and thus not exact).*

Question 4

In this report, we summarize our simulation of individual claims and its implications on year-end assets and risk of bankruptcy. Additionally, we provide recommendations to the company and considerations for future analysis.

4.1 Methodology

Assumptions

To conduct this analysis, we made several assumptions. First, we know the probability of a customer making a claim in any one year is 10%. We assume that the probability of a customer filing a claim is independent of their previous claims and independent of other customers' claims. We also assume that customers do not submit more than one claim per year. Lastly, we assume that the size of the claims are independent and follow a Pareto distribution with parameters *alpha* = 3 and *Beta* = 100,000. The Pareto distribution is suitable for modelling insurance claims because its heavy tail captures the potential for extreme losses.

Reservations

According to Philbrick (1985), two major sources leading to inaccuracy in estimates of expected losses are oversimplified models and misestimated parameters. Several of our assumptions fall into these two categories and may therefore impact the validity of our results. One such assumption is the independence of claims. In reality, events like hurricanes affect many people in close geographical proximity, thereby violating our assumption of independence.

Two other problematic assumptions include the constant probability of making a claim and the assumption that clients only claim once each year. In reality, there is likely a dependent relationship between the number and severity of claims. Several authors (Erhardt, & Czado, 2012; Garrido, Genest, & Schulz, 2015; Shi, Feng, & Ivantsova, 2015) have used more flexible methods to successfully model this relationship across different insurance sectors.

Another assumption that may pose problems is the use of a Pareto distribution with parameters $\boldsymbol{\alpha}$ = 3 and $\boldsymbol{\beta}$ = 100,000. Because simulated insurance claim sizes are highly sensitive to changes in parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, accurate estimation of these parameters is critical.

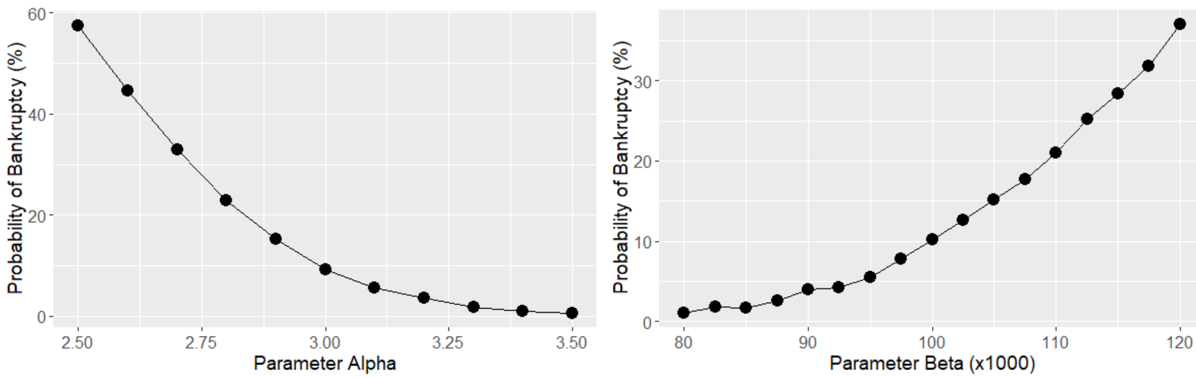

Figure 4.1.1. *Probability of bankruptcy at year-end for varying values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.*

4.2 Results

Using simulation, we investigate how expected assets at year-end and probability of bankruptcy depends on the inputs. First, we find that expected assets at year-end increases as the premium increases, while the probability of bankruptcy decreases (holding all else constant). It is important to note, however, that clients might leave if the premium is set too high.

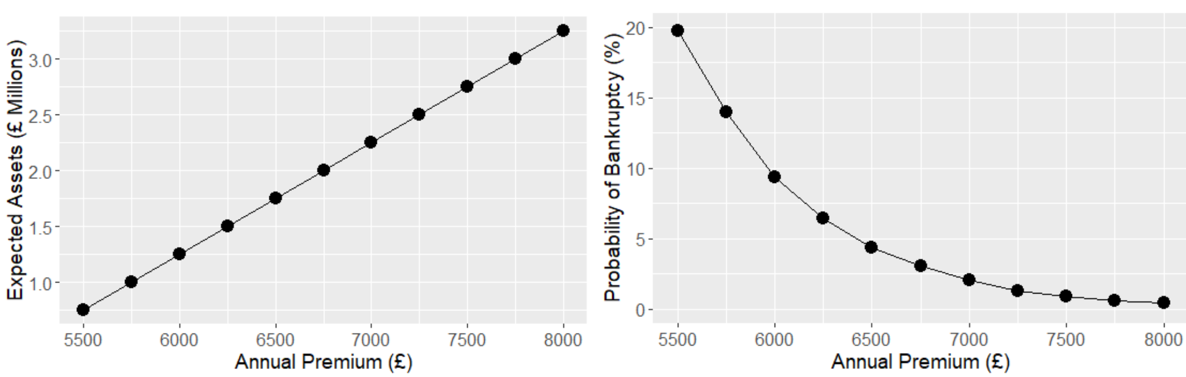

Figure 4.2.1. *Expected assets and probability of bankruptcy at year-end for varying premiums.*

We also explore what happens when the probability of making a claim differs from 10%. When the probability of making a claim increases by 1%-point, expected assets decrease by

£500,000. Additionally, the probability of bankruptcy rapidly increases if the probability of making a claim increases. Therefore, it is essential to reflect on whether the proposed 10% is accurate.

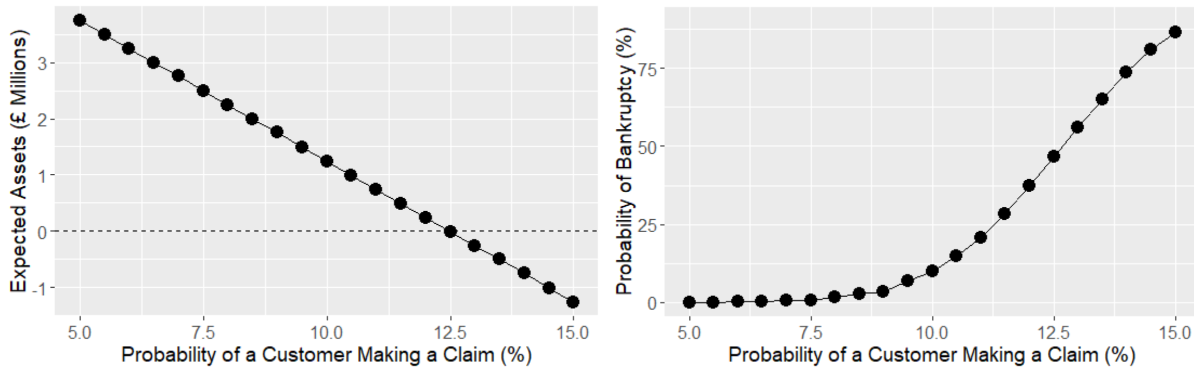

Figure 4.2.2. *Expected assets and probability of bankruptcy at year-end for varying probabilities of a customer making a claim.*

Some more concrete advice is to grow the number of clients. We find that a larger number of clients (with the same characteristics as the current clients) reduces the risk of bankruptcy. The expected profit from a single customer is £1,000, which gives some indication of the maximum amount of money that should be spent on acquiring new clients.

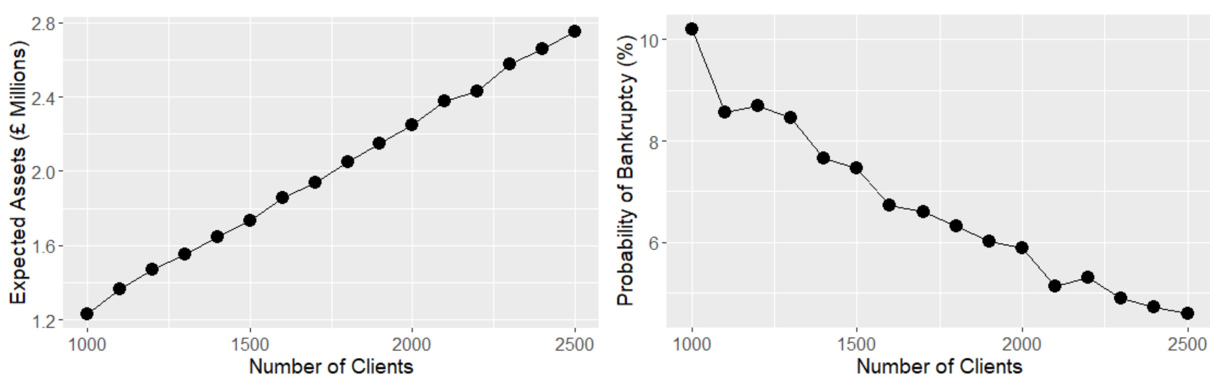

Figure 4.2.3. *Expected assets and probability of bankruptcy at year-end for varying clients.*

We also study how reinsurance can be used to lower the probability of bankruptcy. Reinsurance is a form of insurance purchased by the insurance company itself which (partially) covers any losses. In particular, we study reinsurance which would cover losses up to a certain cap. If the insurance company would like to limit their risk of bankruptcy to 3%, our findings suggest they should insure themselves for losses up to £830,000. A reasonably ‘fair’ price for this insurance would be £26,000.

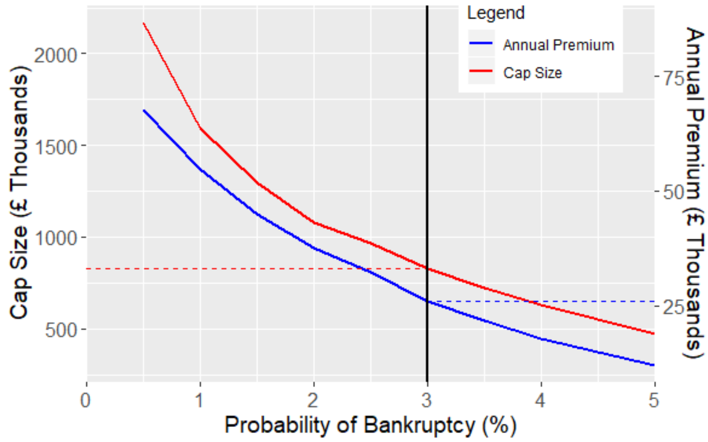

Figure 4.2.4. *Probability of bankruptcy at year-end for varying reinsurance premiums and caps.*

Lastly, we study how the assets would develop over multiple years. In general, the future is bright. The probability of bankruptcy decreases rapidly over the years, so the goal, for now, would be to survive the upcoming year.

Table 4.2.1. Probability of Bankruptcy at Year-End (Given that Firm is in Business at Year Start)

Year of Operation	1	2	3	4	5
Probability of Bankruptcy (%)	9.18%	2.72%	1.15%	0.56%	0.26%

4.3 Discussion

Additional Recommendations

Moral hazard, a common problem in insurance, occurs if clients increase their exposure to risk when the insurance company bears the costs of that risk. According to Vandebroek (1993), bonus-malus systems and deductible policies are two mechanisms which can offset moral hazard. Bonus-malus systems work by adjusting premiums according to a client's individual claims history, thereby rewarding those clients who do not make claims in a given year and punishing those who do. Deductible policies allow clients to face costs up to a certain limit, with the insurance company covering all additional expenses. Both systems align the interests of clients and insurance companies, as both parties bear the costs of the risk.

Another challenge, known as adverse selection, occurs when clients have more information on their behaviour and are thus able to decide whether the insurance benefits them. This information asymmetry creates a division, as only the "bad" clients buy the insurance. Solutions include the introduction of a selection/exclusion policy before selling insurance or setting premiums. Van Winssen, van Kleef, and van de Ven (2018) showed that adverse selection in the Dutch healthcare system could be primarily counteracted by differentiating the premium based on a highly refined risk-rating.

Further Considerations

One possible extension to our analysis is to study the impact on the results when the independence assumptions regarding the claims do not hold. Theoretically, this might be difficult, but in the simulations some dependencies can be introduced.

Furthermore, the simulations can be made more realistic by including the services of a bank. If the insurance firm requires some additional capital at year-end, it could approach the bank for a loan, which would lower the probability of bankruptcy in all cases.

4.4 Executive Summary

Through our analysis, we were able to offer insights into the insurance business. In the upcoming year, there is roughly a 10% probability of bankruptcy, which decreases rapidly in later years. Actions to lower the risk of bankruptcy include increasing the number of clients or purchasing reinsurance.

Appendix A. References

Erhardt, V. & Czado, C., 2012. Modeling dependent yearly claim totals including zero claims in private health insurance. *Scadinavian Actuarial Journal,* 2012(2), pp. 106-129.

Garrido, J., Genest, C., & Schulz, J., 2016. Generalized linear models for dependent frequency and severity of insurance claims. *Insurance: Mathematics and Economics,* 70, pp. 205-215.

Philbrick, S., 1985. A practical guide to the single parameter Pareto distribution. *Proceedings of the Casualty Actuarial Society,* LXXII, pp. 44–84.

Shi, P., Feng, X., & Ivantsova, A., 2015. Dependent frequency–severity modeling of insurance claims. *Insurance: Mathematics and Economics,* 64, pp. 417-428.

Van Winssen, K.P.M., Van Kleef, R.C. & Van de Ven, W.P.M.M., 2018. Can premium differentiation counteract adverse selection in the Dutch supplementary health insurance? A simulation study. *The European Journal of Health Economics,* 19(5), pp. 757-768.

Vandebroek, M., 1993. Bonus-malus system or partial coverage to oppose moral hazard problems? *Insurance: Mathematics and Economics,* 13(1), pp. 1-5.

Appendix B. Additional Analysis

B.1 Long-Run Probability of Bankruptcy

In addition, we consider what happens over a longer period. Note that the number of claims, denoted by C, exhibits $C \sim Binom(1000, p)$ and the value of the claims $X_i \sim Pareto(\alpha, \beta)$. The expected value of total claims is:

$$E\left[\sum_{i=1}^{C} X_i\right] = E\left[E\left[\sum_{i=1}^{C} X_i \mid C\right]\right] = E\left[CE[X_1]\right] = E[C\frac{\beta}{\alpha - 1}] = \frac{\beta}{\alpha - 1}Np$$

where we assume the X_i's are i.i.d. For the given values, the expected value of total claims is £5 million, while the total premium received per year is £6 million. So, the assets are expected to increase by £1 million per year. If a firm survives year 1, it is expected to have more assets at year-end than in the beginning. Therefore, the firm will be less likely to default in the following year. We thus believe the probability of bankruptcy to decrease over the years.

We have done some additional simulations to investigate the probabilities of bankruptcy in later years. In the table below, we give the probability of bankruptcy in some year conditional on the firm being in business at the start of the year.

Table B.1.1. Probability of Bankruptcy at Year End (Given that Firm is in Business at Year Start)

Year of Operation	1	2	3	4	5
Probability of Bankruptcy (%)	9.18%	2.72%	1.15%	0.56%	0.26%

B.2 Number of Clients

Next, we consider in further detail what happens to the probability of bankruptcy when the circumstances change. We define a random variable I_i as an indicator for whether client i made a claim. The indicator I_i takes on values of 1 or 0 with the following probabilities:

$$P(I_i = 1) = p = 0.1,$$
$$P(I_i = 0) = 1 - p = 0.9.$$

We let $X_i$denote the size of the claim which follows the specified Pareto distribution. The "claim" made by client i can then be written as $Y_i = I_i X_i$, where a claim of 0 means no claim is made. For the total value of the claims, we can thus write $\sum_{i=1}^{NC} Y_i$. We have:

$\mu_Y = E[Y_i] \ = p\, E[X_i] \ = \ \frac{\beta}{\alpha - 1} p$ and

$E\left[{Y_i}^2\right] = p\, E\left[{X_i}^2\right] = p\ \frac{2\beta^2}{(\alpha-1)(\alpha-2)}$, so that

$${\sigma_Y}^2 = var(Y_i) \ = \ \frac{2\,\beta^2 p\,(\alpha-1)\ -\ \beta^2 p^2 (\alpha-2)}{(\alpha-1)^2(\alpha-2)}$$

The probability of bankruptcy can then be estimated using the central limit theorem (CLT), where M denotes the premium paid per client:

$$P\left[\sum_{i=1}^{NC} Y_i > NC * M\right] = P\left[\frac{1}{NC}\sum_{i=1}^{NC}(Y_i - \mu_Y)\ > M - \mu_Y\right] =$$

$$P\left[\sqrt{NC}\ \frac{\frac{1}{NC}\sum_{i=1}^{NC} Y_i\ -\ \mu_Y}{\sigma_Y} > \sqrt{NC}\ \frac{M-\ \mu_Y}{\sigma_Y}\right] \approx 1 - \Phi\left(\sqrt{NC}\ \frac{M-\mu_Y}{\sigma_Y}\right)$$

Here, $\Phi(\cdot)$ denotes the standard normal CDF. This estimate can be relatively inaccurate for two reasons. First, the random variables Y_i are mixed random variables with a large probability of being zero and a small probability of being very large. It takes many clients for the mean to be approximately normal. Furthermore, the probability of bankruptcy is contained in the tails of the distribution. For accurate approximations in the tails using the CLT, many observations are required. Despite these concerns, the approximation does suggest that the firm can lower its probability of bankruptcy by increasing the number of clients (as Φ is an increasing function). This is as expected, since the weak law of large numbers suggests that $\frac{1}{NC}\sum_{i=1}^{NC} Y_i \ \to_p\ \mu_y < M$.

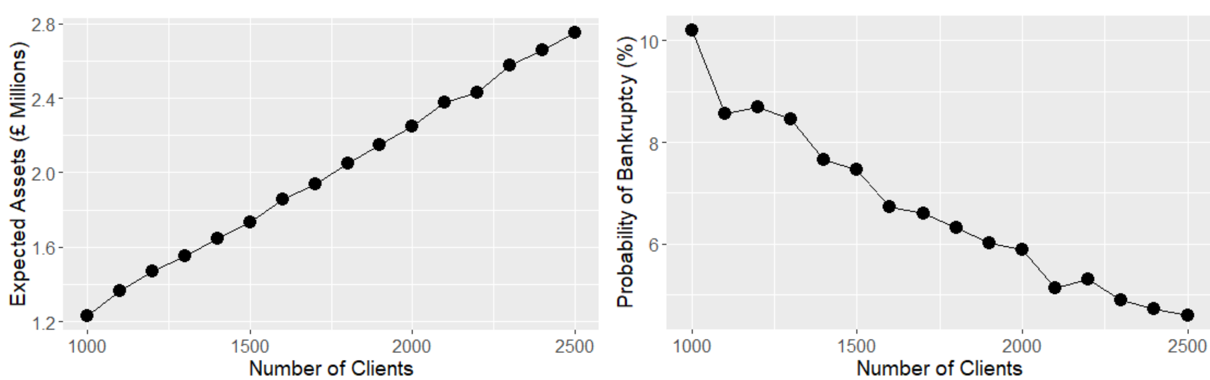

Figure B.2.1. *Left: expected assets (£ millions) at year end for varying numbers of clients. Right: Probability of bankruptcy at year end (%) vs. number of clients. Note: this line is not as smooth as the others since all the claims are simulated again when considering a different number of clients (i.e. when changing the premium, we simply consider the same set of simulated claims and compute the assets at year-end using the various premiums.) Here, the simulations for different number of clients are unrelated.*

B.3 Reinsurance

As a final addition, we study the concept of reinsurance in further detail. Reinsurance refers to when the insurance company purchases insurance from another company to lower their risk of bankruptcy. We consider reinsurance of the following form: reinsurance is purchased for a price *P* and covers any losses up to the amount *L*. If losses at year-end exceed *L*, the company goes bankrupt. In that case, assets at year-end are:

$$assets_1 = assets_0 + NC * premium - total\ claims - P.$$

The probability of bankruptcy is: $P(bankrupt) = P(assets_1 < -L)$.

We assume that the reinsurance premium is of the form:

$P \ = \ -1.1E[costs\ for\ reinsurance\ company] =$

$-1.1\ \{0 * P(assets_1 < -L\ or\ assets_1 > 0) + P(-L \leq assets_1 < 0) * E[A_1| - L \leq assets_1 < \ 0]\}$.

This is reasonable, as the premium should reflect the expected shortfall capped at *L*. The reinsurance company should also be profitable, so we introduce the factor 1.1.

Note that for a given cap L, one can find the corresponding probability of bankruptcy and reinsurance premium. Determining the reinsurance premium is not straightforward. We use an iterative approach where assets at year-end and reinsurance premiums are computed repeatedly until convergence. The plot below displays the cap and corresponding 'fair' price for this reinsurance. For example, to limit its probability of bankruptcy to 3%, the insurance company would need reinsurance that covers losses up to £830,000. A reasonable price for such reinsurance would be £26,000.

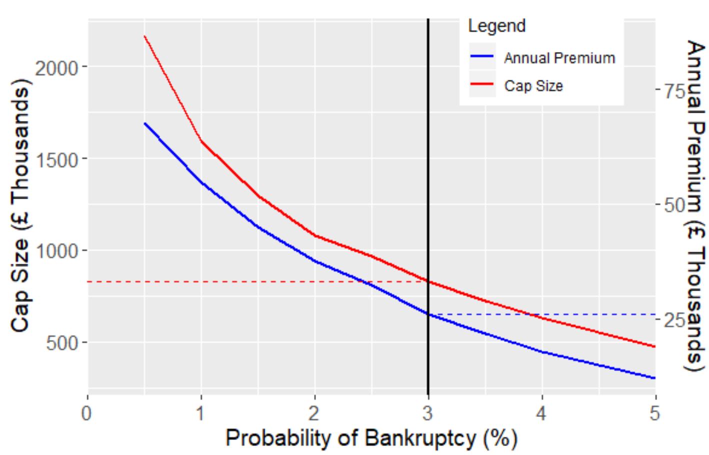

Figure B.3.1. *Probability of bankruptcy (%) at year end for varying annual reinsurance premiums (£ thousands) and cap sizes (£ thousands). The probability of bankruptcy increases as cap size and annual reinsurance premium decrease.*

Appendix C. R Code

```
# Clear working directory
rm(list=ls())

# Install relevant packages (if necessary)
# install.packages("ggplot2")

# Load relevant packages
library(ggplot2)

# Define relevant inputs
init_assets <- 250000
nclient <- 1000
premium <- 6000
probclaim <- 0.1
alpha <- 3
beta <- 100000

# Set seed so that simulation results are replicable
set.seed(1)

############################################################
###################### Question 1C #######################
############################################################

# Write a function to calculate the inverse of the Pareto CDF
invpar <- function(u, a, b) {
  m <- (1 - u) ^ (1 / a)
  x <- (b / m) - b
  return(x)
}

# Write a function that uses the inversion method to simulate
# the values drawn from X
rpar <- function(n, a, b) {

  # Make sure that inputs meet assumptions for alpha, beta
  if(a <= 0 | b <= 0) {
    print("Invalid input")
    return(0)
  }

  # Generate n random values from Unif(0, 1)
  u <- runif(n)

  # Run these values through the 'invpar' function
  x <- invpar(u, a, b)

  return(x)
}

# Simulate 10,000 values drawn from X
n <- 10000
x <- rpar(n, alpha, beta)
```

```r
# Write a function to calculate the true density function
density <- function(z) {
  (alpha * (beta ^ alpha)) / ((z + beta) ^ (alpha + 1))
}

# Get a sense of the distribution of simulated values from X
summary(x)

# Add x to a dataframe so it will work with ggplot
dfr <- data.frame(x)

# NOTE: For aesthetic reasons, we will set upper xlim = 500,000
sum(x<500000)/length(x) # 0.9962
# Our histogram will still include over 99.5% of the observations

# Create ggplot
ggplot(dfr, aes(x = x)) +

  # Add a histogram of 10,000 values drawn from X
  geom_histogram(aes(y = ..density..), bins = 100,
                 colour = "black", fill = "white") +

  # Superimpose the true density function
  stat_function(fun = density, col = "red", size = 1) +

  # Adjust the x-axis limits
  xlim(0, 500000) +

  # Add axis titles
  ylab("Density") +
  xlab("Simulated Insurance Claim Sizes(£)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

###########################################################
####################### Question 2 #######################
###########################################################

### (1) Simulating assets after 1 year

# Write a function to estimate the assets at year end
assets1 <- function(init_assets, premium, nclient, probclaim, a, b) {

  # Simulate for each client a value 0 (no claim) or 1 (claim)
  u = runif(nclient)

  # Calculate the number of claims per year
  nclaim = sapply(probclaim, function(x){sum(u <= x)})

  # Calculate the total value of claims per year
  tot_claims = sapply(nclaim,function(x){sum(rpar(x, a, b))})
```

```
  # Calculate the assets at year end
  assets1 = init_assets + premium * nclient - tot_claims

  return(assets1)
}

# Write a function to generate a large number of assets at year end
MCsimul <- function(B, init_assets, premium, nclient, probclaim, a, b) {

  # Generate the first column of assets at year end
  assets <- assets1(init_assets, premium, nclient, probclaim, a, b)

  # Loop through B-1 more times to calculate assets at year end and
  # add those columns to the 'assets' matrix
  for(i in 2:B){
    assets <- rbind(assets, assets1(init_assets, premium,
                    nclient, probclaim, a, b))
  }

  return(assets)
}

# Simulate assets of the company at year end 10,000 times
B <- 10000
MC <- MCsimul(B, init_assets, premium, nclient, probclaim, alpha, beta)

# Calculate the expected assets at year end
summary(MC / 1000000)

# Calculate the probability of bankruptcy at year end
sum(MC < 0) / B * 100

###########################################################

### (2) Long-run probability of bankruptcy

# Write a function to simulate assets of the company after k years
MCsimul_kyear <- function(B, init_assets, premium, nclient, probclaim, a, b, k){

  # Define starting values/vectors
  assets = init_assets
  pbankrupt <- c()

  # Simulate assets of the company at year end 10,000 times
  assets <- MCsimul(B, init_assets, premium, nclient, probclaim, alpha, beta)

  # Calculate number of firms that went bankrupt after year 1
  bankrupt <- sum(assets < 0)

  # Calculate probability of bankruptcy after year 1
  pbankrupt[1] <- bankrupt / B

  # Calculate number of firms that are still operating after year 1
  nalive <- B - bankrupt

  # Filter out the assets of all bankrupt companies
```

```
  assets <- assets[assets >= 0]

  # Loop through k-1 more times, repeating all the previous steps
  for(i in 2:k){
    MC <- MCsimul(nalive, 0, premium, nclient, probclaim, alpha, beta)
    assets <- assets + MC
    bankrupt <- sum(assets < 0)
    pbankrupt[i] <- bankrupt / nalive
    nalive <- nalive - bankrupt
    assets <- assets[assets >= 0]
  }

  return(list(assets, pbankrupt))
}

# Simulate assets of the company after 5 years
MC5 <- MCsimul_kyear(B, init_assets, premium, nclient, probclaim, alpha, beta, 5)

# Display probability of bankruptcy for years 1 to 5
MC5[[2]]*100

############################################################

### (3) Varying number of clients

# Generate a vector of clients 1000, 1100, ..., 2500
nclients <- seq(1000, 2500, by=100)

# Define empty vectors assets_end and probbankrupt
assets_end <- c()
probbankrupt <- c()

# Simulate assets at year end for each value of nclients
for(i in 1:length(nclients)){

  # Simulate assets of the company at year end 10,000 times
  MC <- MCsimul(B, init_assets, premium, nclients[i], probclaim, alpha, beta)

  # Calculate expected assets at year end and store in vector
  assets_end[i] <- mean(MC) / 1000000

  # Calculate probability of bankruptcy at year end and store in vector
  probbankrupt[i] <- (sum(MC<0)/B) * 100
}

# Add nclients, assets_end and probbankrupt to a dataframe
# so that it is compatible with ggplot
dfr <- data.frame(nclients, assets_end, probbankrupt)

# Create ggplot
ggplot(dfr, aes(x=nclients, y=assets_end)) +

  # Add markers to show bankrupt_premium vs. premiums
  geom_point(size=4) +

  # Connect the markers with a line
```

```
  geom_line() +

  # Add axis titles
  xlab("Number of Clients") +
  ylab("Expected Assets (£ Millions)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12), axis.text.x=element_text(size = 12))

# Create ggplot
ggplot(dfr, aes(x=nclients, y=probbankrupt)) +

  # Add markers to show bankrupt_premium vs. premiums
  geom_point(size=4) +

  # Connect the markers with a line
  geom_line() +

  # Add axis titles
  xlab("Number of Clients") +
  ylab("Probability of Bankruptcy (%)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

###########################################################
###################### Question 3A ######################
###########################################################

# Generate a vector of premiums 5500, 5750, 6000, ..., 8000
premiums <- seq(5500, 8000, by=250)

# Simulate assets of company at year end 10,000 times
MCpremium <- MCsimul(B, init_assets, premiums, nclient, probclaim, alpha, beta)

# Calculate the expected assets at year end
# Divide by 1 million for aesthetic purposes
means_premium <- colMeans(MCpremium) / 1000000

# Calculate the probability of bankruptcy at year end
bankrupt_premium <- vector()
for(i in 1:length(premiums)) {
  bankrupt_premium[i] <- sum(MCpremium[,i] < 0) / nrow(MCpremium) * 100
}

# Add premiums, expected assets and prob. bankruptcy to a
# dataframe so that it is compatible with ggplot
dfr <- data.frame(premiums, means_premium, bankrupt_premium)

# Create ggplot
ggplot(dfr, aes(x = premiums, y = means_premium)) +
```

```
  # Add markers to show expected assets at year end for each premium
  geom_point(size = 4) +

  # Connect the markers with a line
  geom_line() +

  # Add axis titles
  xlab("Annual Premium (£)") +
  ylab("Expected Assets (£ Millions)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

# Create ggplot
ggplot(dfr, aes(x=premiums, y=bankrupt_premium)) +

  # Add markers to show bankrupt_premium vs. premiums
  geom_point(size=4) +

  # Connect the markers with a line
  geom_line() +

  # Adjust the y-axis limits
  ylim(0,20) +

  # Add a horizontal line at prob_bankrupcty = 2%
  geom_hline(yintercept=2, linetype="dashed", colour="red", size=1) +

  # Add an arrow pointing to the optimal premium
  geom_segment(aes(x=7250, y=7.5, xend=7050, yend=3), size=1.3,
               arrow=arrow(length=unit(0.5, "cm"))) +

  # Add a text box
  annotate("label", x=7250, y=10, label="Premium = 7000\nP(Bankruptcy) = 2.07%") +

  # Add axis titles
  xlab("Annual Premium (£)") +
  ylab("Probability of Bankruptcy (%)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

###########################################################
####################### Question 3B ######################
###########################################################

# Generate a vector of claim probs 0.05, 0.055, 0.06, ..., 0.15
probclaims <- seq(0.05, 0.15, by = 0.005)

# Simulate assets of company at year end 10,000 times
```

```r
MCclaims <- MCsimul(B, init_assets, premium, nclient, probclaims, alpha, beta)

# Calculate the expected assets at year end
# Divide by 1 million for aesthetic purposes
means_claims <- colMeans(MCclaims) / 1000000

# Calculate the probability of bankruptcy at year end
bankrupt_claims <- vector()
for(i in 1:length(probclaims)) {
  bankrupt_claims[i] <- sum(MCclaims[,i] < 0) / nrow(MCclaims) * 100
}

# Multiply probclaims by 100 for aesthetic purposes
probclaims <- probclaims * 100

# Add probclaims, means_claims and bankrupt_claims to a dataframe
# so it will work with ggplot
dfr <- data.frame(probclaims, means_claims, bankrupt_claims)

# Create ggplot
ggplot(dfr, aes(x = probclaims, y = means_claims)) +

  # Add markers to show expected assets at year end for each premium
  geom_point(size = 4) +

  # Connect the markers with a line
  geom_line() +

  # Add a horizontal line at y = 0
  geom_hline(yintercept = 0, linetype = "dashed") +

  # Add axis titles
  xlab("Probability of a Customer Making a Claim (%)") +
  ylab("Expected Assets (£ Millions)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

# Create ggplot
ggplot(dfr, aes(x = probclaims, y = bankrupt_claims)) +

  # Add markers to show bankrupt_claims vs. probclaims
  geom_point(size = 4) +

  # Connect the markers with a line
  geom_line() +

  # Adjust the y-axis limits
  ylim(0, 90) +

  # Add a horizontal line at prob_bankrupcty = 2%
  geom_hline(yintercept = 2, linetype = "dashed", colour = "red", size = 1) +
```

```r
  # Add an arrow pointing to the optimal claim probability
  geom_segment(aes(x = 8.3, y = 25, xend = 8.05, yend = 6), size = 1.3,
               arrow = arrow(length = unit(0.5, "cm"))) +

  # Add a text box
  annotate("label", x = 8.3, y = 35,
           label = "P(Claim) = 8%\nP(Bankruptcy) = 1.65%") +

  # Add axis titles
  xlab("Probability of a Customer Making a Claim (%)") +
  ylab("Probability of Bankruptcy (%)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

# Repeat the same analysis for more precise estimate of maximum probability
# of making claim #so that P(bankrupt)<2%
probclaims <- seq(0.08, 0.085, by = 0.0005)
MCclaims <- MCsimul(B, init_assets, premium, nclient, probclaims, alpha, beta)
means_claims <- colMeans(MCclaims) / 1000000
bankrupt_claims <- vector()
for(i in 1:length(probclaims)) {
  bankrupt_claims[i] <- sum(MCclaims[,i] < 0) / nrow(MCclaims) * 100
}
probclaims <- probclaims * 100
rbind(probclaims,bankrupt_claims)

###########################################################
############# Additional Analysis for Report ##############
###########################################################

### (1) Varying alpha

# Generate a vector of alpha's
alpha_seq <- seq(2.5, 3.5, by = 0.1)
df <- data.frame(alpha_seq)

# Compute probability of bankruptcy for each alpha
for(i in 1:length(alpha_seq)) {
  df[i, 2] <- 100 * sum(MCsimul(B, init_assets, premium, nclient,
                      probclaim,alpha_seq[i], beta) < 0) / B
}

colnames(df) <- c("alpha", "pbankruptcy")

# Create ggplot
ggplot(df, aes(x = alpha, y = pbankruptcy)) +

  # Add markers to show alpha vs. pbankruptcy
  geom_point(size = 4) +

  # Connect the markers with a line
  geom_line() +
```

```
  # Add axis titles
  xlab("Parameter Alpha") +
  ylab("Probability of Bankruptcy (%)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

###########################################################

### (2) Varying beta

# Generate a vector of beta's
beta_seq <- seq(80000, 120000, by = 2500)
df <- data.frame(beta_seq / 1000)

# Compute probability of bankruptcy for each beta
for(i in 1:length(beta_seq)) {
  df[i, 2] <- 100 * sum(MCsimul(B, init_assets, premium, nclient,
                          probclaim, alpha, beta_seq[i]) < 0) / B
}

colnames(df) <- c("beta", "pbankruptcy")

# Create ggplot
ggplot(df, aes(x = beta, y = pbankruptcy)) +

  # Add markers to show beta vs. pbankruptcy
  geom_point(size = 4) +

  # Connect markers with a line
  geom_line() +

  # Add axis titles
  xlab("Parameter Beta (x1000)") +
  ylab("Probability of Bankruptcy (%)") +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12))

###########################################################

### (3) Reinsurance

# Redefine input values
init_assets <- 250000
nclient <- 1000
premium <- 6000
probclaim <- 0.1
alpha <- 3
```

```
beta <- 100000

# Simulate assets after 1 year 10,000 times
MC <- MCsimul(B, init_assets, premium, nclient, probclaim, alpha, beta)

# Write a function to find reasonable caps
probbankrupt_cap <- function(dat, p) {
  # For a given p, estimate M such that P(Assets_1 < -M) = p
  df <- sort(dat, decreasing = F)

  ind = 1

  while(ind / B < p){
    ind = ind + 1
  }
  return(-df[ind - 1])
}


# Write a function to determine the reinsurance premium
# and probability of bankruptcy for a given cap
determine_reinspremium <- function(reinsp0, reinsp1, M) {
  iter = 1
  while(abs(reinsp1 - reinsp0) / reinsp0 > 0.0001 & iter < 1000) {
    # Compute premium and assets at year end repeatedly until convergence
    reinsp0 = reinsp1

    # Assets at year end is original minus premium reinsurance
    A1 = MC - reinsp0

    # Compute probability assets between -M and 0
    probM0 = sum((-M <= A1) & (A1 < 0))/B

    # Compute premium = -1.1 E[costs for reinsurer]
    reinsp1 = -1.1 * probM0 * mean(A1[(-M <= A1) & (A1 < 0)])
    iter = iter + 1
  }
  # Compute actual probability of bankruptcy
  A1 = MC - reinsp0
  pbankrupt <- sum((A1 < -M))/B
  return(list(reinsp1, pbankrupt))
}


# Generate a vector of probabilities of bankruptcy
p <- seq(0.005, 0.05, by = 0.005)

df <- matrix(p, ncol = 1)
df <- cbind(df, matrix(0, nrow = nrow(df), ncol = 3))

for(i in 1:nrow(df)) {

  # Determine caps
  cap <- probbankrupt_cap(MC, p[i])

  # Determine 'fair' reinsurance premium and P(bankruptcy)
```

```
  # (The starting values 0.1*cap and 0.2*cap are arbitrary)
  list_premium_probbankrupt <- determine_reinspremium(0.1 * cap, 0.2 * cap, cap)
  premium <- list_premium_probbankrupt[[1]]
  pbankrupt <- list_premium_probbankrupt[[2]]

  df[i, 2] <- cap / 1000

  df[i, 3] <- premium / 1000

  df[i,4] <- pbankrupt*100
}

# Add elements to dataframe to make them compatible with ggplot
dfr <- data.frame(df)
colnames(dfr) <- c("p", "cap", "premium", "pbankrupt")
dfr$pbankrupt <- dfr$p * 100

# Create ggplot
ggplot(dfr, aes(x = pbankrupt)) +

  # Add a line representing pbankrupt vs. cap
  geom_line(aes(y = cap, colour = "Cap Size"), size=1) +

  # Add a line representing pbankrupt vs. premium
  geom_line(aes(y = premium * 25, colour = "Annual Premium"), size=1) +

  # Put premiums on a secondary axis
  scale_y_continuous(sec.axis =
                     sec_axis(~./25, name =  "Annual Premium (£ Thousands)")) +

  geom_segment(aes(x = 0, y = dfr$cap[6], xend = 3, yend = dfr$cap[6], colour = "Cap Size"),
               linetype="dashed") +

  geom_segment(aes(x = 3, y = dfr$premium[6]*25, xend = 5, yend = dfr$premium[6]*25,
                   colour = "Annual Premium"), linetype="dashed") +

  geom_vline(xintercept=3, size=0.8) +

  # Change the color of the lines
  scale_colour_manual(values = c("blue", "red")) +

  # Add axis labels
  ylab("Cap Size (£ Thousands)") +
  xlab("Probability of Bankruptcy (%)") +
  labs(colour = "Legend") +

  # Move legend to the top of the figure
  theme(legend.position = c(0.8, 0.9)) +

  # Adjust font sizes
  theme(axis.title.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.text.y=element_text(size = 12),
        axis.text.x=element_text(size = 12)) +

  scale_x_continuous(expand = c(0, 0))
```