End-term project: Advanced Statistical Computing 2020

Yizhen Dai *s2395479*

This report is the result of the efforts for the final exam/project of Advanced Statistical Computating. The goal of this project is to analyze the cost and the benefit of a re-insruance policy through simulation.

Keywords: Advanced Statistical Computating, Final Project, Simulation

Introduction

The goal

This project aims at solving a modeling problem faced by an insurance company - ANV. Two of ANV business lines, Professional liability insurance (PLI) and Workers' compensation (WC), were affected by a huge claim from one client during the last year. Therefore, ANV comes to a reinsurance company for an insurance policy. To put the problem in a mathematical format, we define the following notation:

- X_1: the loss incurred for PLI from ANV clients (in million euros);
- X_2: the loss incurred for WC from ANV clients (in million euros);
- t: the threshold set by the reinsurance company (in million euros). For some threshold $t = 100, 110, \ldots, 200$, if the total loss incurred for PLI and WC from a certain client exceeds t, ANV pays the claim themselves; Otherwise, the reinsurance company pays the claim.
- $V(t)$: the expected over-threshold claims, which is equal to $E[(X_1 + X_2)1(X_1 + X_2 > t)]$;
- $P(t)$: the price that reinsurance company asks. $P(t)$ depends on the threshold t as $P(t) = 40000 * e^{-t/7}$.

Our goal is to determine if ANV should buy such policy from the reinsurance company. Of course, the policy is only reasonable if the expected over-threshold claim exceeds the price: $V(t) > P(t)$. The data available is the loss incurred for PLI and WC for all the clients of ANV during last year. With limited data available, we use statistical modeling to approximate $V(t)$ and therefore determine if $V(t) > P(t)$.

Models

X_1 and X_2 are positively correlated. The correlation between X_1 and X_2 is 0.528. The linear regression line (in red) and the *LOWESS* smoother line (in blue) is shown below in Figure 1.

In order to reflect the dependence in the data, we cannot simulate X_1 and X_2 separately. We use a Copula model to model the joint density f_{X_1,X_2}.

$$f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1)\, f_{X_2}(x_2)\, c(u_1, u_2)$$

where $u_1 = F_{X_1}(x_1), u2 = F_{X_2}(x_2)$, which are the marginal functions of f_{X_1} and f_{X_2}; $c(u_1, u_2)$ is the joint integral transforms $U_1 = F_{X_1}(X_1)$ and $U_2 = F_{X_2}(X_2)$.

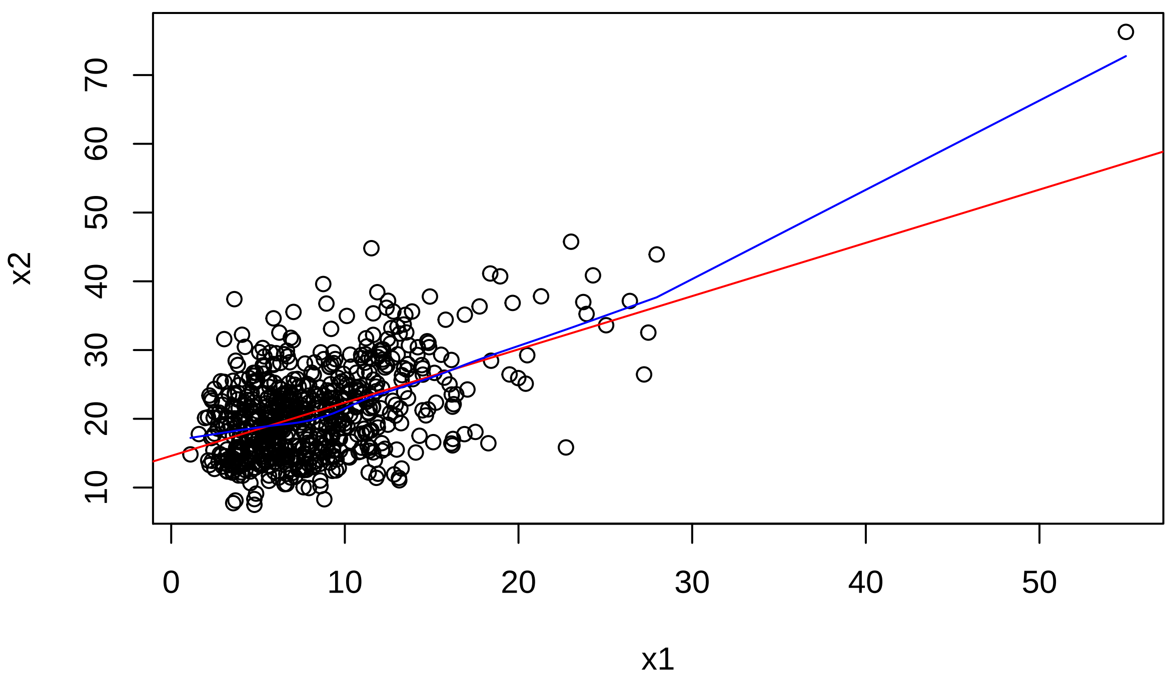

Figure 1: The relationship between X1 and X2

Preliminary experiments suggested the following parametric models:
$f_{X_1}(\cdot;\mu_1,\sigma_1) \sim \text{Lognormal}(\mu_1,\sigma_1), \mu_1 \in \mathbb{R}, \sigma_1 > 0$
$f_{X_2}(\cdot;\mu_2,\sigma_2) \sim \text{Lognormal}(\mu_2,\sigma_2), \mu_2 \in \mathbb{R}, \sigma_2 > 0$
$c(\cdot;\theta) \sim \text{Joe}(\theta), \theta \geq 1$
where a Lognormal distribution is for a random variable whose logarithm is normally distributed; a Joe distribution is one of the most prominent bivariate Archimedean copulas in Copula models.

Methodolgy

Parameter Estimates

Based on the models, we have a total of five parameters to be estimated from the data: $\mu_1, \sigma_1, \mu_2, \sigma_2, \theta$. The maximum likelihood estimation is used to estimate these parameters. The idea behind is to estimate these five parameters by maximizing likelihood function.

First, we estimate $\mu_1, \sigma_1, \mu_2, \sigma_2$ by finding the values that can maximize the sum of log density of the observed data so the data is most probable under our model assumption f_{X_1}, f_{X_2}. To speed up the optimization, we can use the sample mean and sample standard deviation of the observed data on the log scale.
$\hat{\mu} = \arg\max_{\mu} \{\ln \{\prod f_X(\mu \mid x)\}\}$
$\hat{\sigma} = \arg\max_{\sigma} \{\ln \{\prod f_X(\sigma \mid x)\}\}$
Afterwards, we estimate θ using the probability integral transforms u_1, u_2 of the observed data x_1, x_2 using the cumulative distribution function of Lognormal distribution F_{X_1}, F_{X_2}. Such transforms are not observed get pseudo-observations by plugging in the estimated $\mu_1, \sigma_1, \mu_2, \sigma_2$. It would work if our estimated parameters are close to the true parameters.
$u_1 = F_{X_1}(x_1 \mid \hat{\mu_1}, \hat{\sigma_1})$
$u_2 = F_{X_2}(x_2 \mid \hat{\mu_2}, \hat{\sigma_2})$
$\hat{\theta} = \arg\max_{\theta} \{\ln \{\prod_i c(\theta \mid u_1, u_2)\}\}$
The resulted estimated parameters are $\hat{\mu_1} = 1.982, \hat{\sigma_1} = 0.513, \hat{\mu_2} = 2.997, \hat{\sigma_2} = 0.311, \hat{\theta} = 1.608$.

Simulated data

We get the parameter estimates, then we can simulate enough data to get the average cost $V(t)$ over the years.

To generate enough data, we need a function that simulates from the joint model for (X_1, X_2) for a given set of parameters $(\mu_1, \sigma_1, \mu_2, \sigma_2, \theta)$. We firstly simulate u_1, u_2 using the density function of Joe. Then we get x_1, x_2 based on Lognormal distributions.

To evaluate our estimated model, we can compare the simulated data with the observed data. They are shown in Figure 2 and Figure 3.

We can see the marginal and joint distributions of observed data and simulated data are similar, which is a good sign of our implementation.

We can also check change in data distributions when we tune the parameters using our simulation function. Currently $\hat{\mu_1}, = 1.982, \hat{\sigma_1} = 0.513, \hat{\mu_2} = 2.997, \hat{\sigma_2} = 0.311, \hat{\theta} = 1.608$.

First, we increase θ to 4. The results are shown in Figure 4. We can see with a bigger theta, the data are more correlated with smaller deviation from the linear regression line.

Next, we check the distribution with bigger $\mu_1 = 3, \mu_2 = 4$. The results are shown in Figure 5. We can see when we increase μ_1, μ_2, the center of the contour also shifts in the same direction.

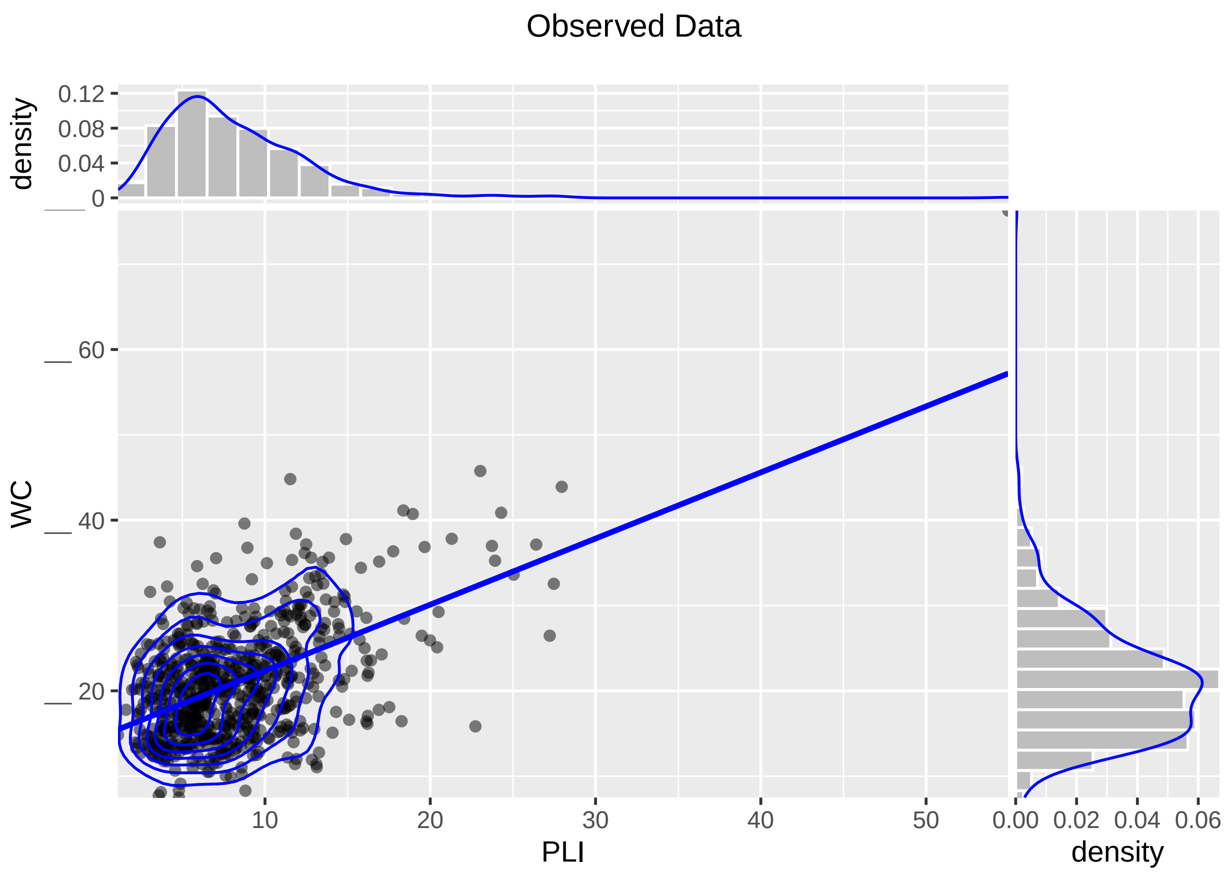

Figure 2: Observed Data

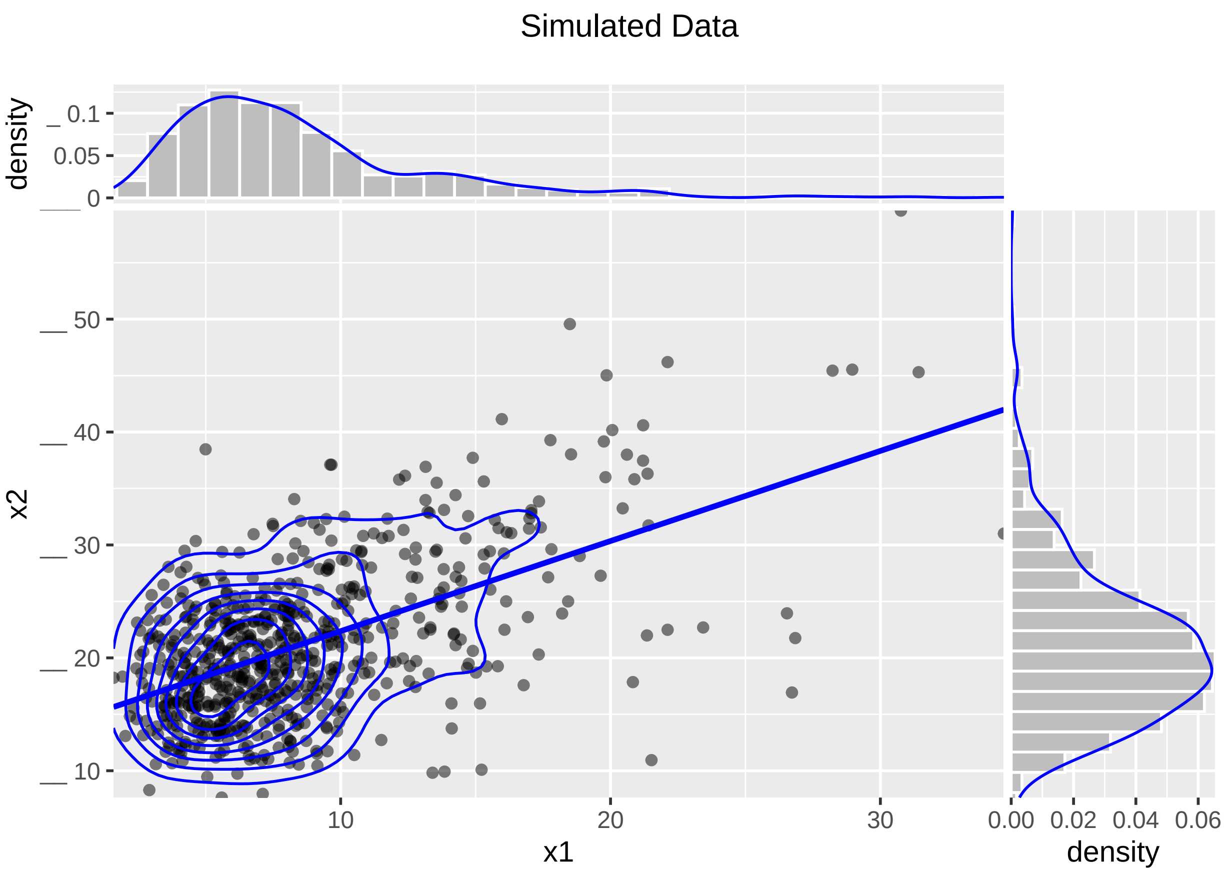

Figure 3: Simulated Data

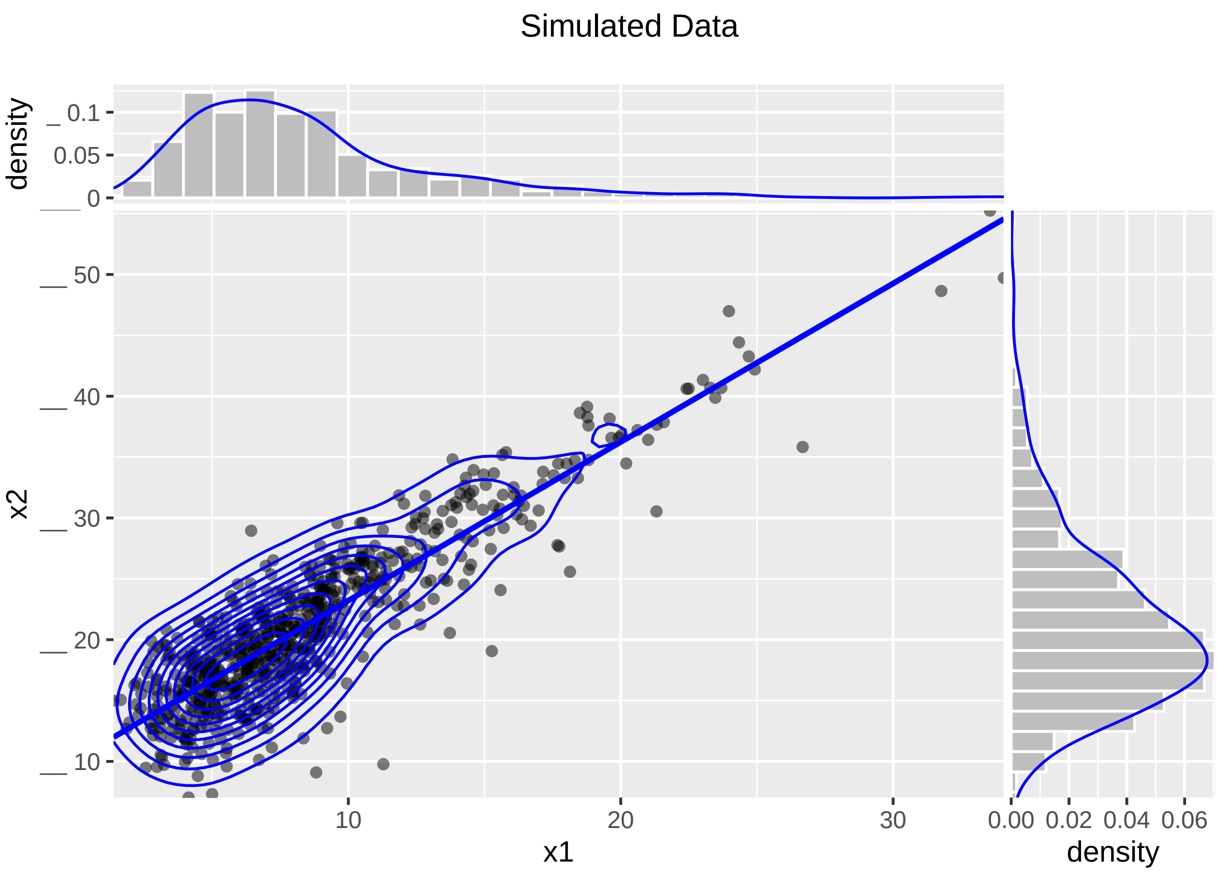

Figure 4: theta = 4

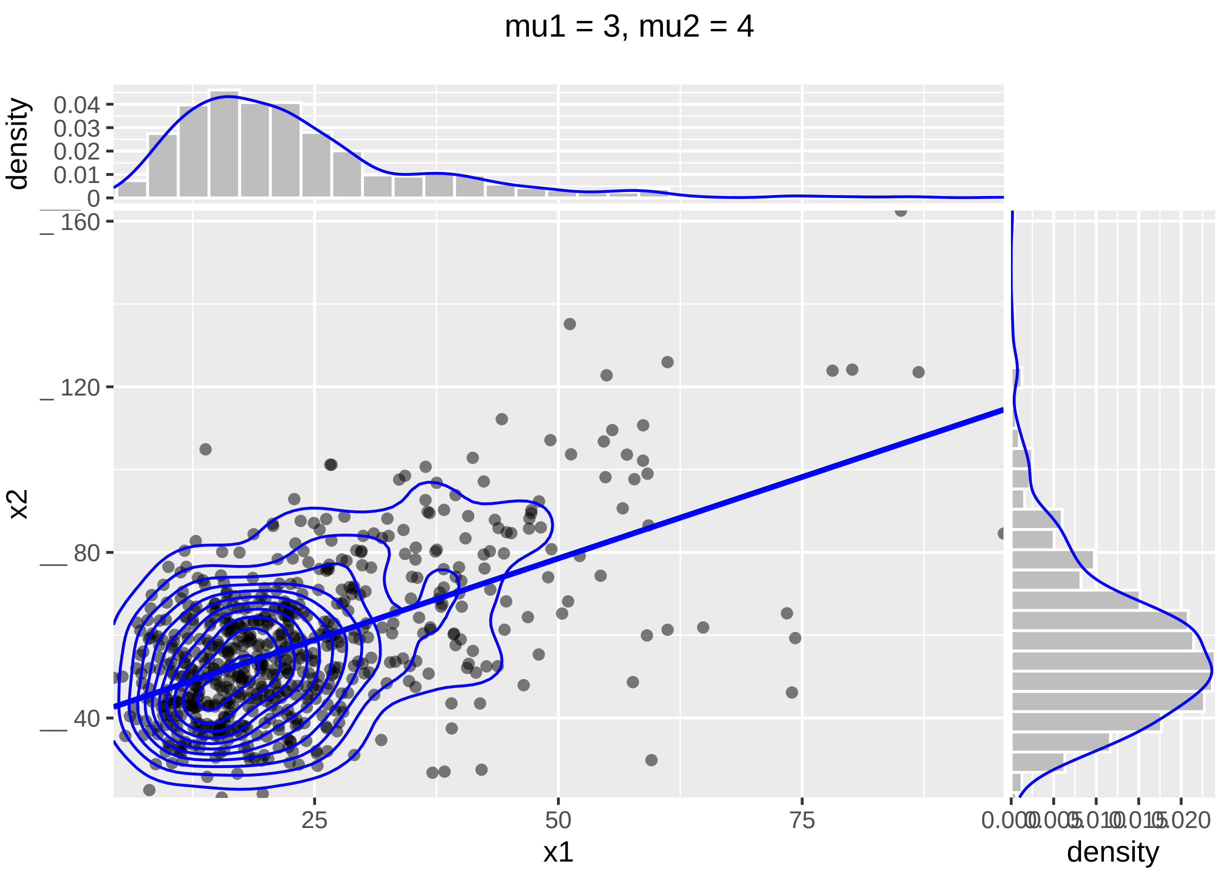

Figure 5: mu1 = 3, mu2 = 4

Next, we check the distribution with bigger standard deviation of Lognormal distribution: $\sigma_1 = \sigma_2 = 0.8$. The results are shown in Figure 6. We can see that the data is more dispersed, which is same as expected. We can also observe that a small change in σ_1, σ_2 has a big influence on the extreme values. This is partly because the σ is on the log scale.

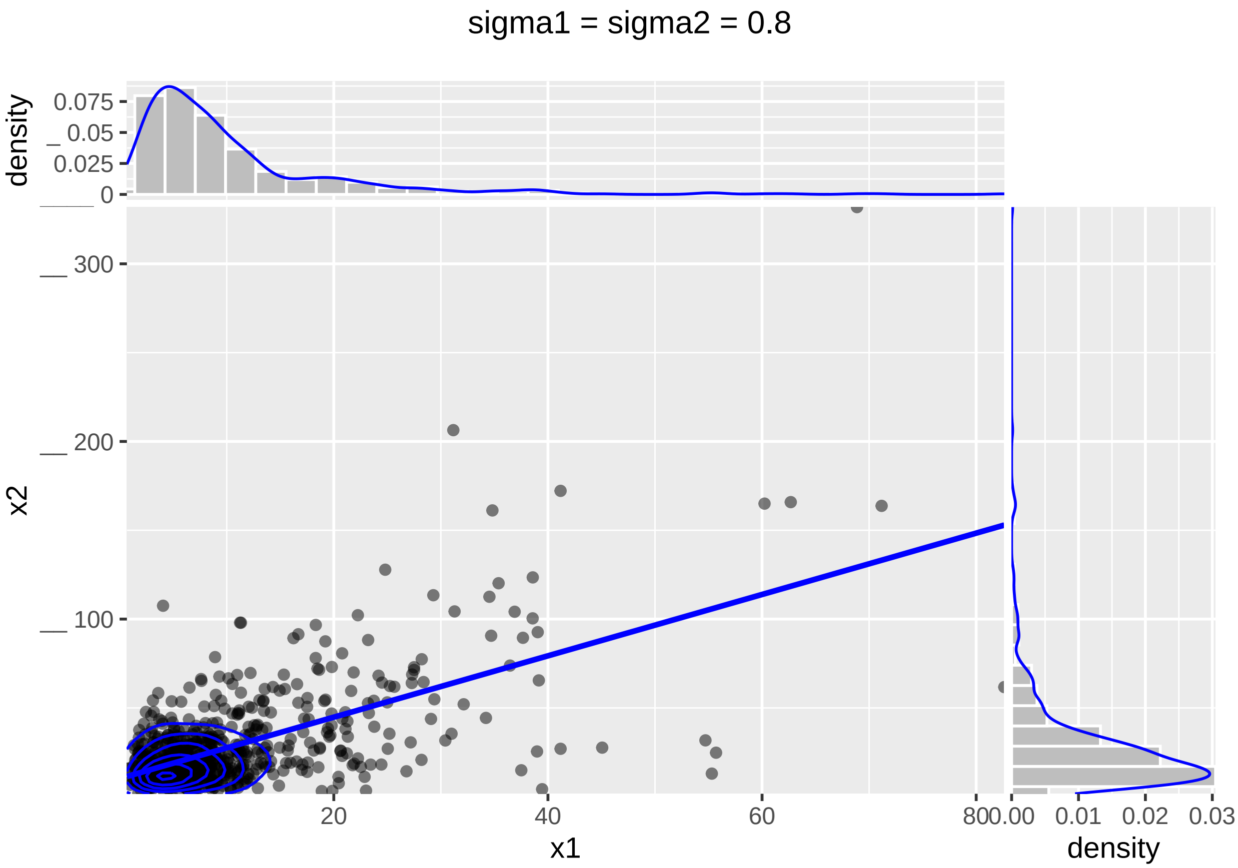

Figure 6: sigma1 = sigma2 = 0.8

Simulation Study

Acuracy evaluation

To evaluate how accurate our parameter estimation can be, we conduct a simulation study assuming the following values for true parameters: $\mu_1 = 1, \sigma_1 = 2, \mu_2 = 3, \sigma_2 = 0.5, \theta = 2$.

We compute the RMSE for 100 simulation runs of 200,500 and 1000 data points.

The average computing time and RMSE are shown in Figure 7 and Figure 8. We can see that required computation time increase linearly with the number of runs. RMSE gets smaller with more simulation runs. θ is the hardest parameter to estimate with largest RMSE. μ_1, σ_1 have higher RMSE than μ_2, σ_2 separately. This is probably due to a lower standard deviation for f_{X_2} (0.5 compared to 2 on the log scale). The distribution for f_{X_1} is more dispersed, leading to a larger RMSE for μ_1, σ_1.

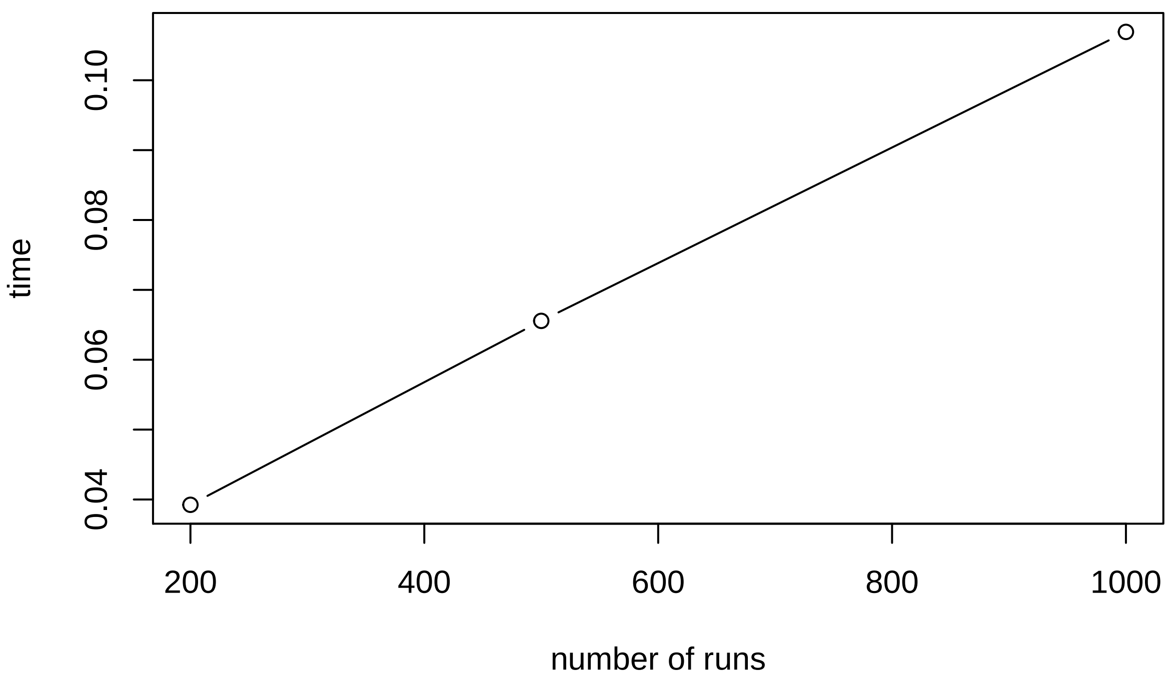

Figure 7: Average computing time vs number of runs

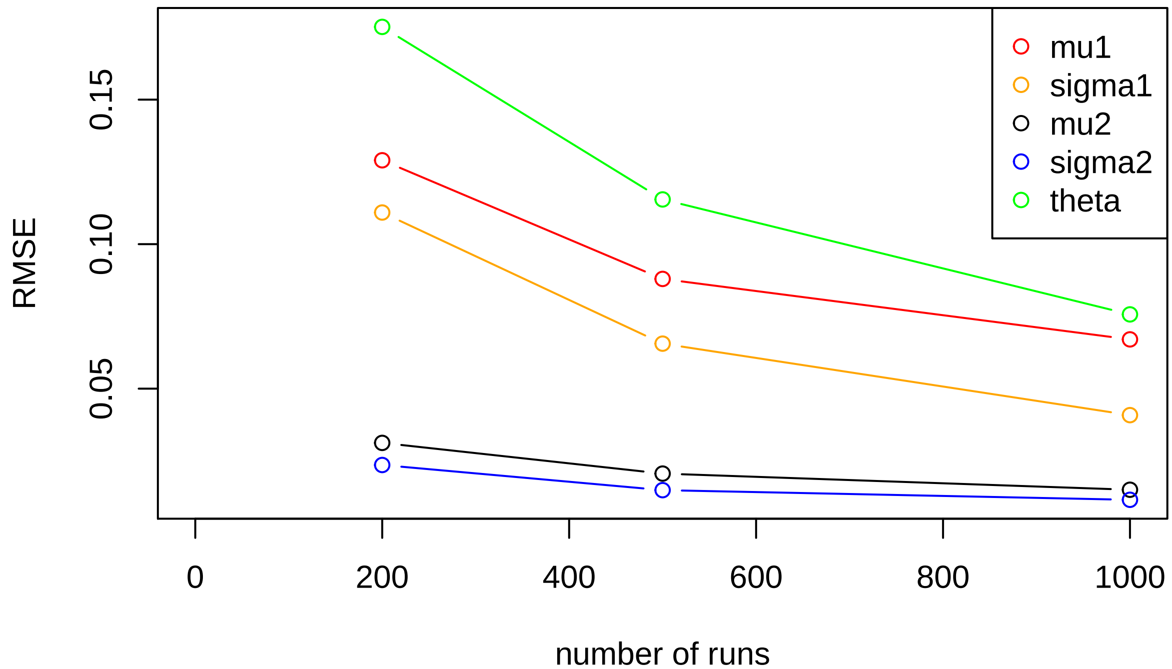

Figure 8: RMSE vs number of runs

Working on the real data: V(t)

Having a feeling of the simulation models, we fit all the five parameters to the observed data. We compute the expected payout of the reinsurance $V(t)$ using parametric Monte Carlo simulation (based on the estimated parameters). We use 10^5 Monte Carlo samples and calculate expected $V(t), t = 100, 110, \ldots, 200$, as the average over the simulation runs.

The plot of $V(t)$ for $t = 100, 110, \ldots, 200$ based on 10^5 Monte Carlo samples is shown in Figure 9.

[1] "P(t) and V(t) based on 10^5 simulutation samples"

	P(t)	V(t)	Std for V(t)	P(t) < V(t)
100	0.0249950	0.0305661	1.8318015	1
110	0.0059901	0.0108184	1.1414797	1
120	0.0014355	0.0074188	0.9579313	1
130	0.0003440	0.0000000	0.0000000	0
140	0.0000824	0.0000000	0.0000000	0
150	0.0000198	0.0000000	0.0000000	0
160	0.0000047	0.0000000	0.0000000	0
170	0.0000011	0.0000000	0.0000000	0
180	0.0000003	0.0000000	0.0000000	0
190	0.0000001	0.0000000	0.0000000	0
200	0.0000000	0.0000000	0.0000000	0

Based on the results, buying the reinsurance policy seems a good idea when $t \leq 120$ since the policy price ($P(t)$)is smaller than the potential cost of not buying the policy ($V(t)$). However, the standard deviation is high for $V(t)$. This is because the claim exceeding t is a highly improbable event.

Importance sampling

To handle the highly improbable events, we must get more samples of claims exceeding t. Therefore, importance sampling comes to play. It is implemented by sampling from an alternative distribution that has higher densities in the region of our interest. In our case, the alternative density function is the Copula function with new $u_1^{\text{new}}, u_2^{\text{new}}$. $u_1^{\text{new}} = \hat{u}_1 + 1, u_2^{\text{new}} = \hat{u}_2 + 1$. We adjust our estimate using an adjustment factor, which is the ratio of the alternative distribution density and the original distribution density.

The plot of $V(t)$ for $t = 100, 110, \ldots, 200$ based on 10^5 Monte Carlo samples and importance sampling is shown in Figure 10. The $V(t)$ based on importance sampling have values when $t >$ 120. Based on the results, the company should buy the reinsurance policy regardless of the value of t since $P(t) < V(t)$.

[1] "P(t) and V(t) based on importance sampling"

	P(t)	V(t)	Std of V(t)	P(t) < V(t)
100	0.0249950	0.0267185	0.2083507	1
110	0.0059901	0.0117026	0.1244900	1
120	0.0014355	0.0054778	0.0782729	1
130	0.0003440	0.0026791	0.0502820	1

	P(t)	V(t)	Std of V(t)	P(t) < V(t)
140	0.0000824	0.0012372	0.0318883	1
150	0.0000198	0.0005347	0.0187541	1
160	0.0000047	0.0002972	0.0128362	1
170	0.0000011	0.0001426	0.0084551	1
180	0.0000003	0.0000781	0.0062483	1
190	0.0000001	0.0000463	0.0047873	1
200	0.0000000	0.0000201	0.0029700	1

Confidence Intervals

Since we are not sure how confident we are about our estimates, we can to conduct bootstrapping to get confidence intervals of $V(t)$. The codes doing a bootstrap method (1000 times) to compute 80% confidence intervals for $V(t)$ are shown below. After getting 1000 results, we can calculate the confidence intervals by getting the 10th and 90th percentiles. The Confidence Intervals for $V(t)$ are shown in Figure 11.

[1] "P(t) and the 90% confidence interval of V(t)"

	P(t)	Lower bound of CI(V(t))	P(t) < V(t)_Lower
100	0.0249950	0.0155202	0
110	0.0059901	0.0064640	1
120	0.0014355	0.0027785	1
130	0.0003440	0.0012229	1
140	0.0000824	0.0005514	1
150	0.0000198	0.0002572	1
160	0.0000047	0.0001214	1
170	0.0000011	0.0000591	1
180	0.0000003	0.0000291	1
190	0.0000001	0.0000142	1
200	0.0000000	0.0000066	1

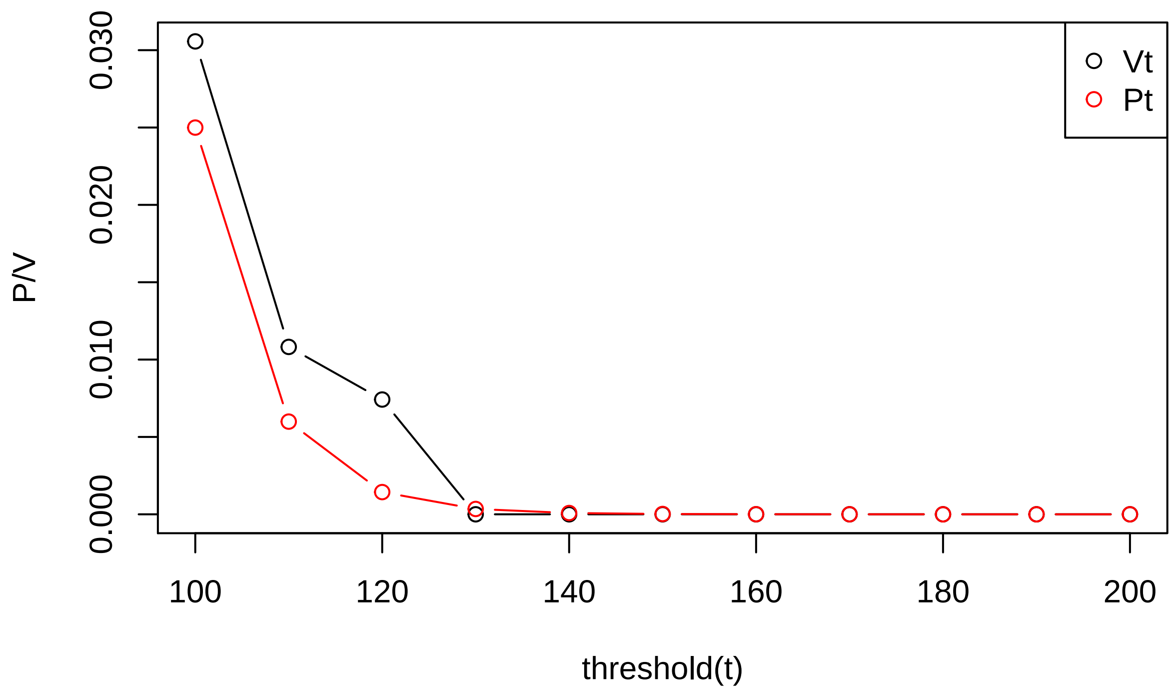

Figure 9: P(t) vs t

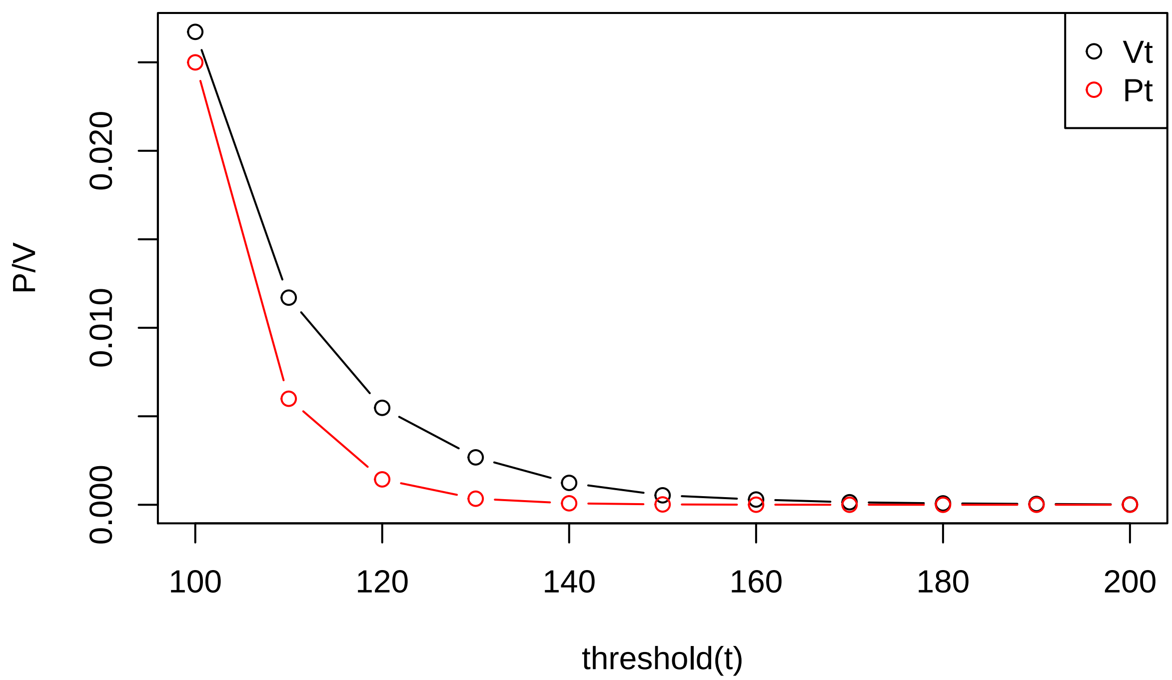

Figure 10: P(t) and V(t) vs t based on importance sampling

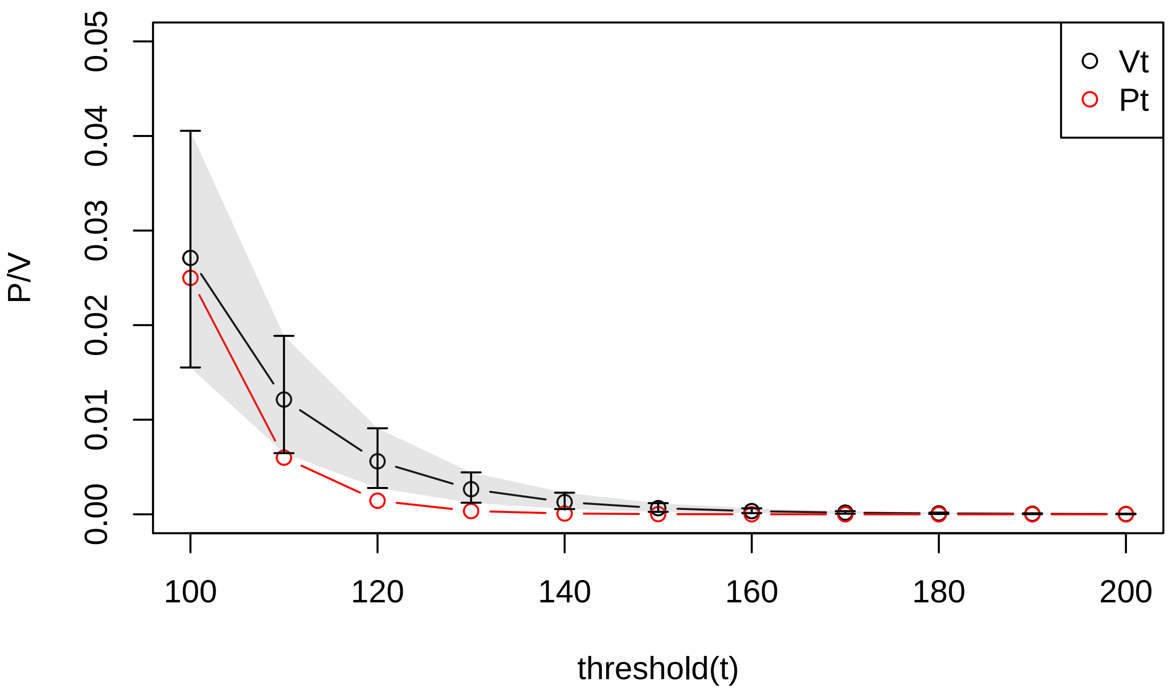

Figure 11: 90% Confidence Interval P(t) in dashed line and V(t)

Results

The implementation does account for Monte Carlo approximation error since we are doing bootstrapping. However, it does not account for estimation error since we are relying on the existing data (648 data points).

As seen from the Figure 11, there is huge uncertainty in the values of P when t is 100. We cannot draw a conclusion that V will be smaller than P on a 90% confidence level when t is 100. However, when $t \geq 110$, the P is smaller than the lower bound of the confidence interval V, indicating that the company ANV should buy the re-insurance.