Universiteit Leiden
The Netherlands

End-term project Advanced Statistical Computing

Solving a Real Life Insurance Problem using Statistical Computing

Author:
Justin Kraaijenbrink

Student number:
s2577984

Special acknowledgements to Dr. T.W. Nagler and C. Schmeits for their help and clear explanations during this project.

October 23, 2020

Introduction

Insurance companies are institutions that provide insurances to people. The risk such companies carry can sometimes exceed a certain threshold, where the expected payout is larger than the revenue. In such cases, the insurance company might turn to a reinsurance company, where the risk is transferred to another party. ANV is such an insurance company. Since they have a profit motive, they surely want to maximize their earnings. However, they encountered some huge claims in two of their business lines: 1. Professional liability insurance (PLI) and 2. Workers' compensation (WC), caused by one client. In order to prevent the company from such huge claims in the future, ANV wants to investigate the possibilities to transfer their risk to a reinsurance company. They have already thought about the specific form of the policy: if PLI + WC $\leq t$ (with t some threshold), ANV pays the claim, otherwise the reinsurance company pays for it. The reinsurance company sets the price for this policy depending on t: $P(t) = 40000 \exp(-t/7)$ (in million euros). It is obvious that the price that ANV pays to the reinsurance company must certainly not exceed the money they save by taking out a policy with the reinsurance company. In other words: the expected payout must be larger than the price paid for the policy. That this should work for ANV can be seen by the law of large numbers: the more samples we generate, the closer the average is to the expectation. Or: the more clients ANV has, the closer the average claim moves towards the expected claim. The expected value of the claim can be expressed as $V(t) = \mathbb{E}[(PLI + WC)\mathbb{1}(PLI + WC > t)]$, with $V(t)$ in million euros. The expected value of the sum of two *independent* random variables is equal to the sum of the expected values of these variables. However, this axiom does not hold when there is dependence in the data. As can be observed in Figure 1, the variables `PLI` and `WC` are moderately correlated ($\rho = .528$). The red point in the upper right corner of

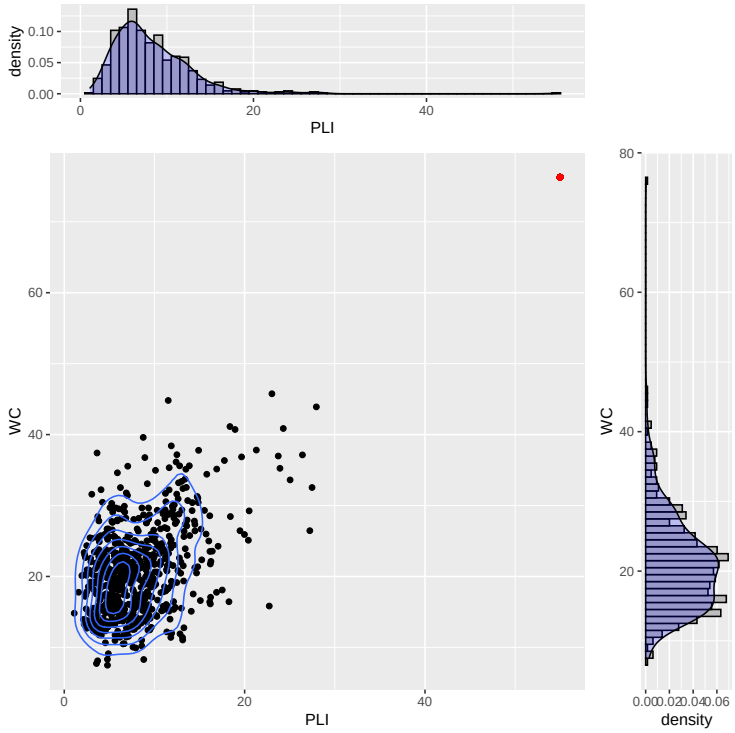

Figure 1: Scatterplot and histograms of the distribution of PLI and WC.

Figure 1 indicates the problematic client. It slightly strengthens the relationship between `PLI` and `WC`, but even without that client the dependence structure would definitively be present. We must therefore turn to numerical methods and statistical modeling to obtain $V(t)$.

Before we can do so, we should translate the problem to more mathematical expressions. To approximate $V(t) = \mathbb{E}[(PLI + WC)\mathbb{1}(PLI + WC > t)]$, we will model the joint distribution F_{X_1,X_2} where $X_1 = PLI$ and $X_2 = WC$. We can use that the joint probability density function f_{X_1,X_2} can be decomposed as follows: $f_{X_1,X_2} = f_{X_1}(x_1)f_{X_2}(x_2)c(F_{X_1}(x_1), F_{X_2}(x_2))$, with:

- $f_{X_1}(\cdot;\mu_1,\sigma_1) \sim \text{Lognormal}(\mu_1,\sigma_1), \mu_1 \in \mathbb{R}, \sigma_1 > 0$
- $f_{X_2}(\cdot;\mu_2,\sigma_2) \sim \text{Lognormal}(\mu_2,\sigma_2), \mu_2 \in \mathbb{R}, \sigma_2 > 0$
- $c(\cdot;\theta) \sim \text{Joe}(\theta), \theta \geq 1$

with c called the *copula density*, which is the joint density of the probability integral transforms $U_1 = F_{X_1}(X_1)$ and $U_2 = F_{X_2}(X_2)$. Note the clever use of the following theorem:

$$\text{If } U \sim \text{Unif}(0,1) \text{ and } X = F^{-1}(U), \text{ then } X \sim F$$

From Figure 1 it should be clear that the distributions of X_1 and X_2 are indeed Lognormal. By now, the model is fully specified and important assumptions are made and justified, so we can move on to the methodology on how to approach $V(t)$.

Methodology

In order to obtain $V(t)$, the following steps were performed:

1. Write a function that uses Maximum Likelihood to estimate the model parameters $\mu_1,\ \sigma_1,\ \mu_2,\ \sigma_2$ and θ
2. Write a function that simulates data from the joint model for a given set of parameters
3. Conduct a simulation study to see which parameters are easier or harder to estimate in terms of RMSE
4. Use Monte Carlo (MC) simulation methods to estimate the expected payout $V(t)$
5. Perform an empirical bootstrap to compute 80% confidence intervals for $V(t)$

Two important remarks are in place here. First of all, Maximum Likelihood is eminently suitable to obtain parameter estimates for $\mu_1,\ \sigma_1,\ \mu_2,\ \sigma_2$ and θ, since it is developed to find the parameter values that are most likely *given the observed data*. The likelihood can be expressed as $L(\theta;x) = \prod_{i=1}^{n} f(X_i;\theta)$. It is common practice to take the log of this expression, because it will transform the product into a sum. Taking the derivative of that sum with respect to a particluar parameter and equating to zero yields the ML estimate for that parameter. For the Lognormal-densities of X_1 and X_2 we have analytic expressions for μ and σ available, but it is also possible to use numerical methods. Here we used R's built-in function `optim()` to approximate the ML estimates. `optim()` takes as input the starting values for the parameters to be estimated, a (likelihood) function and some data. For μ_1 and μ_2, we used as starting parameters the mean of $\log(X_1)$ and $\log(X_2)$, respectively. For σ_1 and σ_2 we used $\sqrt{\sum(\log(X_1) - \hat{\mu_1})^2/N}$ and

$\sqrt{\sum(\log(X_2) - \hat{\mu_2})^2}/N$, respectively, with $\hat{\mu}$ the starting value for μ. Furthermore, θ was initially set to 1.

A second remark concerns the fitting of θ of the Joe copula model. We use *pseudo-observations* $\hat{U}_j = F_{\hat{\mu_j},\hat{\sigma_j}}(X_j), j = 1, 2$. As was already mentioned in the introduction, this makes clever use of the theorem that if $U \sim \text{Unif}(0, 1)$ and $X = F^{-1}$, then $X \sim F$. If that is true, then the reverse must also hold: if $X \sim F$, then $F(X) = U \sim \text{Unif}(0, 1)$. Now let's move on to the actual simulations!

Simulation study and Results

The function that uses maximum likelihood to obtain parameter estimates yielded the following values: $\mu_1 = 1.982$, $\sigma_1 = 0.513$, $\mu_2 = 2.997$, $\sigma_2 = 0.311$ and $\theta = 1.608$. R's built-in function `dlnorm()` was used to for the likelihood function, as well as the function `dCopula()` from the `copula`-package (Hofert M, Kojadinovic I, Maechler M, Yan J (2020). copula: Multivariate Dependence with Copulas. R package version 1.0-0, `https://CRAN.R-project.org/package=copula`). Figure 2 presents $n = 648$ simulated data points from the joint distribution with MLE parameters, as well as the true data points. The two data sets look fairly similar!

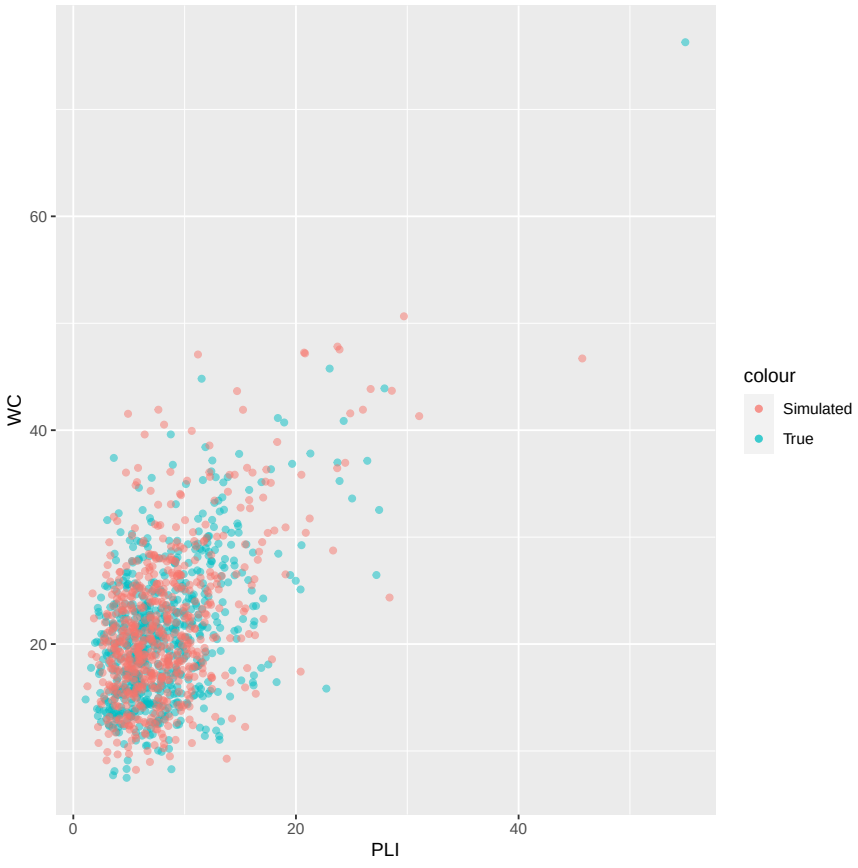

Figure 2: Simulated data points and true data points.

The properties of the simulated data change according to the parameter values. Figure 3 illustrates what happens if we adjust μ_1, σ_1 and θ. The red points are simulated data points for μ_1 increased by

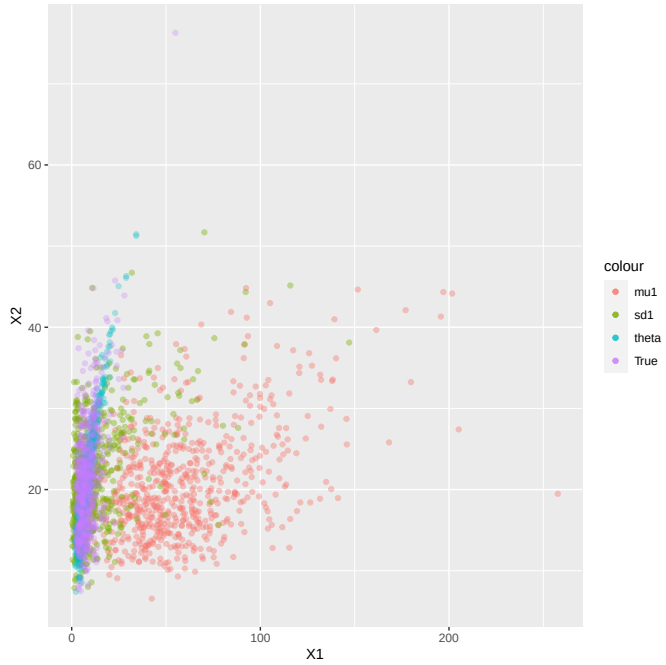

Figure 3: Simulated data points with adapted parameter values for μ_1, σ_1 and θ.

factor 2. We see that there is much more spread in the direction of X_1 compared to the purple points, which represent the true data. A similar pattern is observed for doubling σ_1. If we would have repeated this for μ_2 and σ_2, the spread would have been in the direction of X_2. The blue points represent simulated data points where θ has been quadrupled. Larger θ implies a stronger correlation between X_1 and X_2, which is evident from the fact that the points lie more on a line compared to the true data.

To get more insight into the inner workings of this parameter estimation, we implemented a simulation study as follows:

Algorithm 1 Simulation study parameter estimation

1: **for all** $n \in \{100, 200, 500, 1000\}$ **do**
2: Simulate n data points
3: Estimate parameters $\mu_1,\ \sigma_1,\ \mu_2,\ \sigma_2,\ \theta$ based on simulated data with fixed parameters $\mu_1 = 1,\ \sigma_1 = 2,\ \mu_2 = 3,\ \sigma_2 = 0.5,\ \theta = 2$
4: Repeat steps 2. and 3. $r = 100$ times
5: **for** $\mu_1,\ \sigma_1,\ \mu_2,\ \sigma_2,\ \theta$ **do**
6: Calculate $RMSE = \sigma/r$ with σ the standard deviation of the $r = 100$ parameter estimates
7: **end for**
8: **end for**

Figure 4 presents the RMSE for each parameter as a function of the number of simulated data points n.

We see that for all parameters the RMSE decreases, which we could have expected since the RMSE depends on the sample size. We also see that θ has the largest RMSE, followed by μ_1 and μ_2. The higher the RMSE, the 'harder' a parameter is to estimate. θ is in this sense the most difficult parameter to

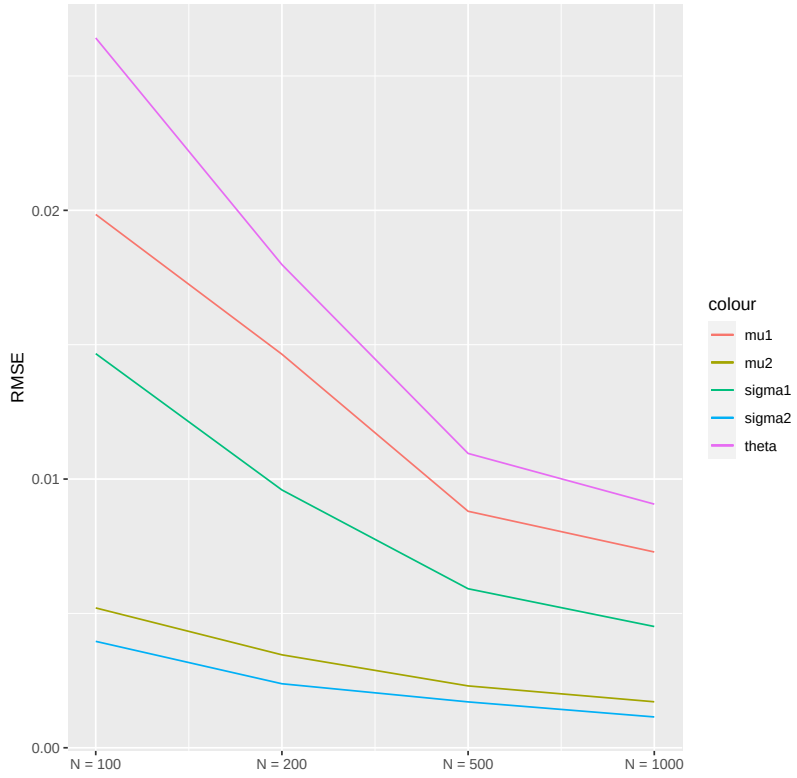

Figure 4: RMSE for $n = \{100, 200, 500, 1000\}$ and each parameter.

estimate accurately, which makes perfect sense because it depends on both X_1 *and* X_2. Whereas the μ's and σ's are concerned with simulation fluctuations in only one variable, θ suffers from simulation inaccuracy of both variables.

Figure 5 shows the average computing times as function of n for estimating all five parameter values. The computing times are decreasing in n. This can be explained from the fact that simulating for example twice as many data points is not twice as expensive in terms of runtime.

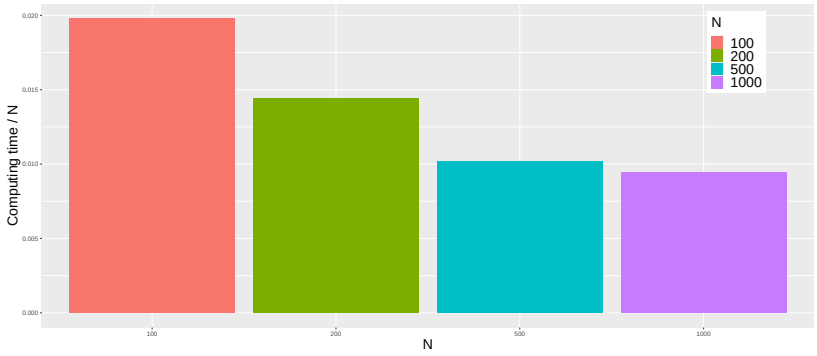

Figure 5: Average computing times for $n = \{100, 200, 500, 1000\}$.

Monte Carlo simulation To estimate the expected payout $V(t)$ we used Monte Carlo simulation. A first approach was to use plain MC, where we simulated 10^5 points from the joint distribution and calculated $(PLI + WC)\mathbb{1}(PLI + WC > t)$ for each datapoint. The mean of these calculated values is an estimate of the expected payout $V(t)$. This process was executed for $t = \{100, 110, ..., 200\}$. Results are presented in Figure 6.

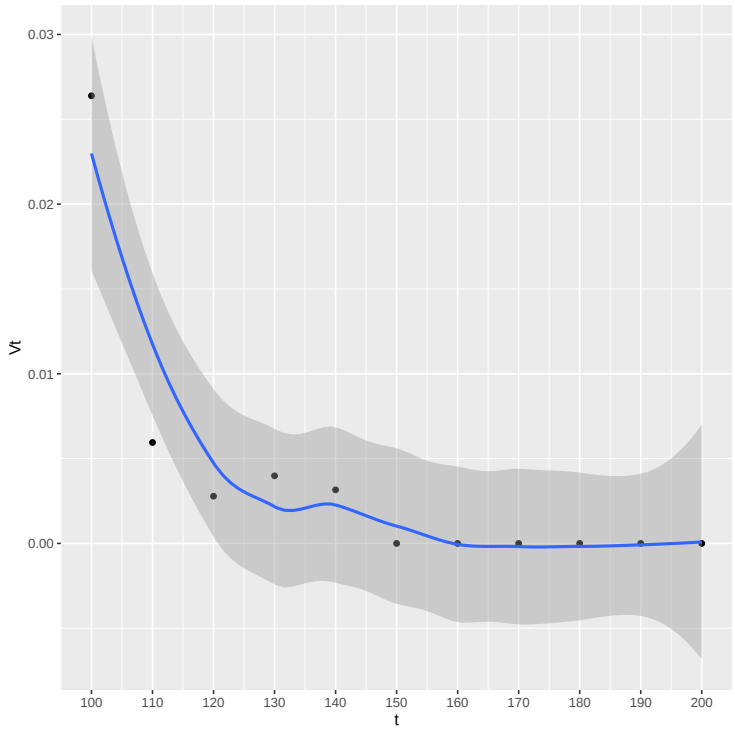

Figure 6: Plain MC estimates for $V(t)$ with $t = \{100, 110, ..., 200\}$.

What cannot be observed from Figure 6 is that the estimates are quite instable and rather noisy. This is due to the fact that $(PLI + WC > t)$ is highly improbable. Actually, there is only one (!) observation for which the expression is true for $t < 140$. When $t \geq 140$, there are no observations for which the expression holds true. A better approach would be to use importance sampling:

Algorithm 2 Importance sampling

1: Simulate $N = 10^5$ data points from the joint distributuion with new parameters, such that $(PLI + WC > t)$ becomes more likely. In our simulation study, this was done by increasing μ_1 with 1 and leaving the other parameters untouched
2: Obtain weights for importance sampling by dividing the Lognormal-density of the simulated data with the old parameters (MLE) by the Lognormal-density of the simulated data with the new parameters (using `dlnorm()`).
3: **for all** $t \in \{100, 110, ..., 200\}$ **do**
4: Calculate $\frac{1}{N}\sum_{i=1}^{N}((PLI_i + WC_i)\mathbb{1}(PLI_i + WC_i > t))$
5: **end for**

Results are presented in Figure 7, where the price P as function of t is also plotted.

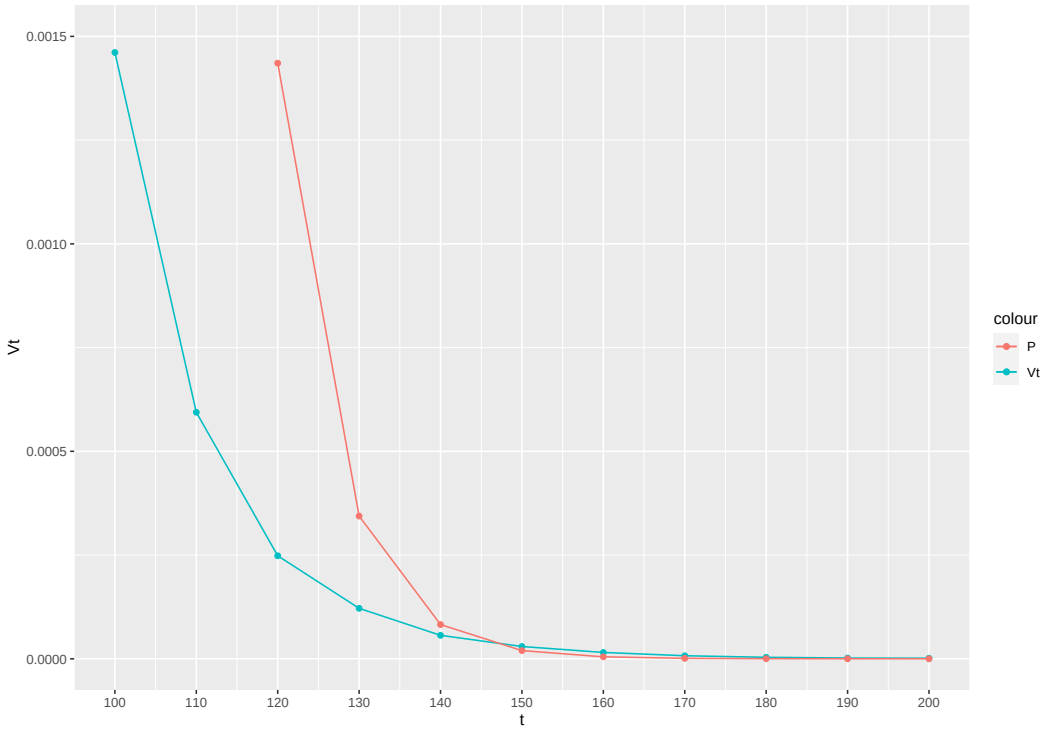

Figure 7: MC importance sampling results for $V(t)$ and $P(t)$ plotted against t.

From $t = 150$ onwards, the expected payout exceeds the policy price, so from these results we may conclude that ANV would benefit from an insurance policy for $t \geq 150$. However, these results are based on a single simulation of 10^5 data points, but as Advanced Statistical Computationalists we could do better! That is why the last step in this simulation study was to perform an empirical bootstrap, so that we could estimate 80% confidence intervals for $V(t)$. The following procedure was followed:

Algorithm 3 Bootstrap procedure

1: **for** 1:B, with B the number of bootstrap replicates **do**
2: Create a bootstrap sample by sampling $n = 648$ observations with replacement from the true data
3: Estimate ML parameters based on the bootstrap sample
4: Create new parameters by adding 1 to μ_1
5: **for all** $t \in \{100, 110, ..., 200\}$ **do**
6: Perform importance sampling to obtain $V(t)$
7: **end for**
8: **end for**
9: Calculate the lower and upper bound of the basic bootstrapped confidence interval $(2\hat{\theta} - \hat{q}_{.90}, 2\hat{\theta} - \hat{q}_{.10})$, with $\hat{\theta}$ the bootstrapped mean for $V(t)$ and $\hat{q}_\alpha$ the empirical α-quantile of the bootstrapped values for $V(t)$.

Figure 8 shows a similar plot as Figure 7, but now with the bootstrap-based confidence intervals added.

An important remark must be made that concerns the estimation of $V(t)$. We sampled only $B = 1000$

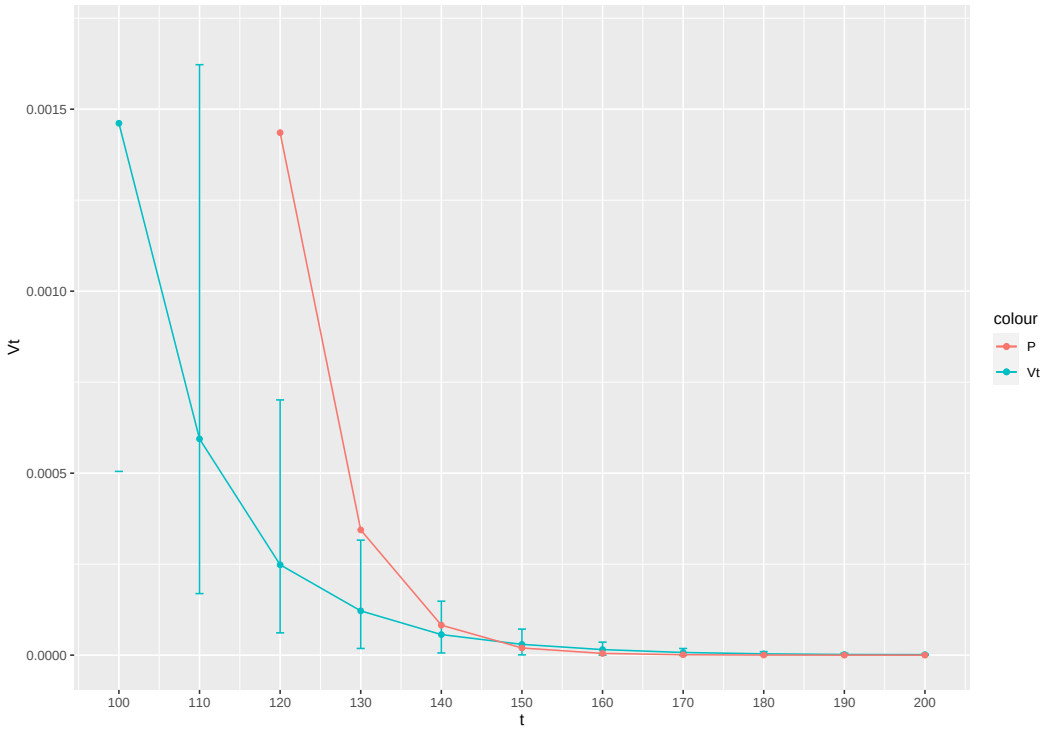

Figure 8: MC importance sampling results for $V(t)$ and $P(t)$, including 80% CIs for $V(t)$.

times from the original data, and calculate $V(t)$ for each t based on the the same sample. However, we could also have sampled *for each* t $B = 1000$ times. The first approach yields dependence across t between the confidence intervals of $V(t)$, but the latter latter slows down runtime significantly. Since the dependence in t would only cause problems when there are a lot of extreme cases (and there are not), we chose to prefer runtime improvement.

We see that we must be cautious with our previous conclusion that ANV should buy the reinsurance policy for $t \geq 150$. For example, when $t = 140$ the price $P(140)$ lies within the confidence interval of $V(140)$. For $t = 150$ this applies as well, so we cannot be confident that for $t \geq 150$ the expected payout is indeed higher than the policy price. Note that 80% confidence intervals should be interpreted as follows: 'when we repeat the experiment 1000 times, the true parameter lies within the interval 800 times'. The bootstrap does account for estimation error, since we assume that the bootstrap mean is equal to the 'true' value. However, this is not the same as the MC approximation error, which is defined as $\frac{\sigma}{\sqrt{N}}$, where σ is the standard deviation of the parameter estimates and N is the number of replicates. The bootstrap does not explicitly take this uncertainty into account, although the approximation errors are in some sense 'averaged' over the bootstrap samples.

Conclusion

In this report we used statistical modeling and simulation study to solve a complex insurance problem. We used maximum likelihood to estimate model parameters of the joint distribution, implemented a simulation study to get more insight in the underlying mechanisms of our parameter estimation method

and used Monte Carlo simulation to obtain estimates for the expected payout $V(t)$. To investigate the robustness of our estimates for $V(t)$ using 80% confidence intervals, an empirical bootstrap was executed.

We found that we can be quite sure that for $t \leq 130$ ANV should not buy the reinsurance policy. However, for $t \geq 140$ decisions should be made very carefully, since the results are somewhat more ambiguous. For example, for $t = \{140, 150\}$ the expected payout can exceed the policy price, but the reverse may also be true. For $t \geq 160$ we are a bit more confident that the expected payout is larger than the policy price, since the price lies close to the lower bound of the confidence interval of $V(160)$. For $t \geq 170$, the policy price lies consistently below the lower bound of the confidence interval of $V(t)$.

We hope that this simulation study will help ANV to make proper decisions concerning their insurance politics!