ETH Zürich, D-MATH
HS 2018
Prof. Dr. Mario V. Wüthrich

Coordinator
Andrea Gabrielli

Non-Life Insurance: Mathematics and Statistics

Solution sheet 7

Solution 7.1 Hill Estimator

An example of a possible R code is given below. The Hill plot (on the left) and the log-log plot (on the right) are given in Figure 1. Note that even though we sampled from a Pareto distribution with tail index $\alpha = 2$, it is not at all clear to see that the data comes from a Pareto distribution. In the Hill plot we see that, first, the estimates of α seem more or less correct, but starting from the 180 largest observations, the plot suggests a higher α or even another distribution. In the log-log plot we see that for small-sized and medium-sized claims the fit seems to be fine. But looking at the largest claims, we would conclude that our data is not as heavy-tailed as a true Pareto distribution with threshold $\theta = 10$ million and tail index $\alpha = 2$ would suggest. We are confronted with these problems even though we sampled directly from a Pareto distribution. This might indicate the difficulties one faces when trying to fit such a distribution to a real data set, where, to make matters even worse, we often have far less than 300 observations (as we have in this example) and, moreover, the observations may be contaminated by other distributions.

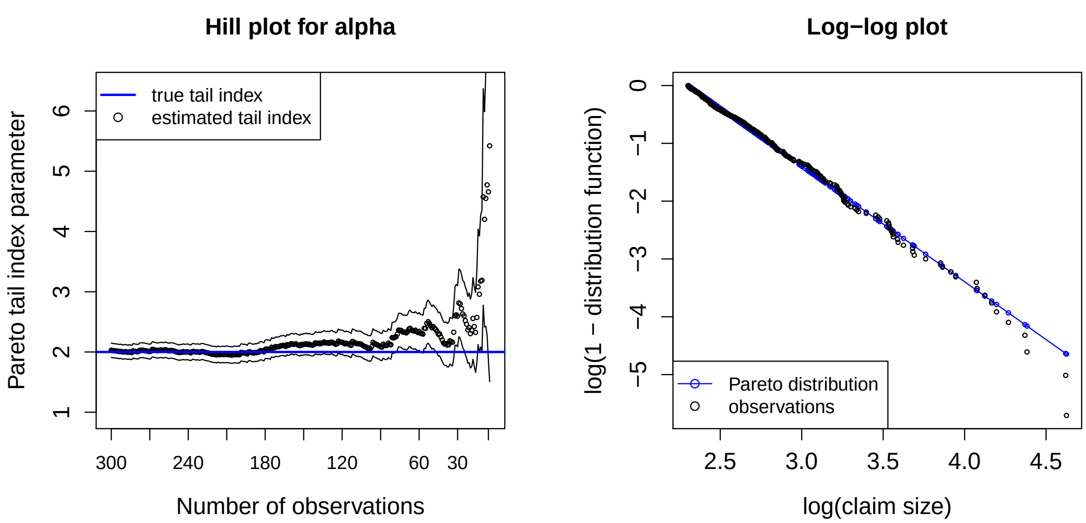

Figure 1: Hill plot for determining the tail index α (left). Log-log plot for the observations and the Pareto distribution (right).

```
### Define the function that creates the Hill plot
hill.plot.function <- function(n, theta, alpha, seed1){

  ### Generate n independent Pareto observations
  set.seed(seed1)
  data.1 <- rgamma(n, shape = 1, scale = 1/alpha)
  data <- theta * exp(data.1)

```

```
  ### Order the data
  data.ordered <- data[order(data, decreasing = FALSE)]

  ### Take the logarithm
  log.data.ordered <- log(data.ordered)

  ### Number of observations
  n.obs <- n:5

  ### Hill estimator
  hill.estimator <- ((sum(log.data.ordered)-cumsum(log.data.ordered
     ) + log.data.ordered)[-((n-3):n)]/n.obs - log.data.ordered[-((
     n-3):n)])^(-1)

  ### Confidence bounds (see Lemma 3.8 of the lecture notes)
  upper.bound <- hill.estimator + sqrt(n.obs^2/((n.obs-1)^2*(n.obs
     -2))*hill.estimator^2)
  lower.bound <- hill.estimator - sqrt(n.obs^2/((n.obs-1)^2*(n.obs
     -2))*hill.estimator^2)

  ### Hill plot
  plot(hill.estimator, ylim = c(min(hill.estimator)-1,max(hill.
     estimator)+1), xaxt="n", xlab = "Number of observations", ylab
      = "Pareto tail index parameter", main = "Hill plot for alpha"
     , cex = 0.5, cex.lab = 1.25, cex.main = 1.25, cex.axis = 1.25)
  axis(1,at=c(1,seq(from = n/10+1, to = n*9/10+1, by = n/10),n-5),
     c(seq(from = n, to = n/10, by = -n/10),5))
  lines(upper.bound)
  lines(lower.bound)
  abline(h = alpha, col = "blue",lwd=2)
  legend("topleft", col=c("blue", "black"), lty=c(1,NA), pch = c(NA
     ,1), lwd=c(2,NA), legend=c("true tail index","estimated tail
     index"))
}

### Define the function that creates the log-log plot
log.log.plot.function <- function(n, theta, alpha, seed1){

  ### Generate n independent Pareto observations
  set.seed(seed1)
  data.1 <- rgamma(n, shape = 1, scale = 1/alpha)
  data <- theta * exp(data.1)

  ### Order the data and take the logarithm
  log.data.ordered <- log(data[order(data, decreasing = FALSE)])

  ### True survival function
  true.sf <- (data.ordered/theta)^(-alpha)

  ### Empirical survival function
  empirical.sf <- 1 - (1:n)/(n+1)

```

```
  ### Log-log plot
  plot(log.data.ordered, log(true.sf), xlab = "log(claim size)", ylab = "log(1 - distribution function)", ylim = c(min(log(true.sf),log(empirical.sf)),max(log(true.sf),log(empirical.sf))), main = "Log-log plot", cex.lab = 1.25, cex.main = 1.25, cex.axis = 1.25, cex = 0.5, col = "blue")
  lines(log.data.ordered,log(true.sf), col = "blue")
  points(log.data.ordered, log(empirical.sf), col = "black", cex=0.5)
  legend("bottomleft", col=c("blue", "black"), lty=c(1,NA), pch = c(1,1), legend=c("Pareto distribution","observations"))
}

### Apply the function for the Hill plot with the desired parameters
hill.plot.function(n=300,theta=10,alpha=2,seed1=100)

### Apply the function for the log-log plot with the desired parameters
log.log.plot.function(n=300,theta=10,alpha=2,seed1=100)
```

Solution 7.2 Approximations for Compound Distributions

Note that if $Y \sim \Gamma(\gamma = 100, c = \frac{1}{10})$, then

$$\begin{aligned}
\mathbb{E}[Y] &= \frac{\gamma}{c} = \frac{100}{1/10} = 1'000,\\
\mathbb{E}[Y^2] &= \frac{\gamma(\gamma+1)}{c^2} = \frac{100 \cdot 101}{1/100} = 1'010'000 \quad \text{and}\\
\mathbb{E}[Y^3] &= \frac{\gamma(\gamma+1)(\gamma+2)}{c^3} = \frac{100 \cdot 101 \cdot 102}{1/1000} = 1'030'200'000.
\end{aligned}$$

Let M_Y denote the moment generating function of Y. According to formula (1.3) of the lecture notes, we have

$$M_Y'''(0) = \frac{d^3}{dr^3} M_Y(r)\bigg|_{r=0} = \mathbb{E}[Y^3].$$

For the total claim amount S, we can use Proposition 2.11 of the lecture notes to get

$$\begin{aligned}
\mathbb{E}[S] &= \lambda v \mathbb{E}[Y] = 1'000 \cdot 1'000 = 1'000'000,\\
\operatorname{Var}(S) &= \lambda v \mathbb{E}[Y^2] = 1'000 \cdot 1'010'000 = 1'010'000'000 \quad \text{and}\\
M_S(r) &= \exp\{\lambda v [M_Y(r) - 1]\}.
\end{aligned}$$

In order to get the skewness ς_S of S, which we will need for the translated gamma and the log-normal approximations, we can use the third equation given in the formulas (1.5) of the lecture notes:

$$\varsigma_S \cdot \operatorname{Var}(S)^{3/2} = \frac{d^3}{dr^3} \log M_S(r)\bigg|_{r=0} = \lambda v \frac{d^3}{dr^3} M_Y(r)\bigg|_{r=0} = \lambda v M_Y'''(0) = \lambda v \mathbb{E}[Y^3],$$

from which we can conclude that

$$\varsigma_S = \frac{\lambda v \mathbb{E}[Y^3]}{(\lambda v \mathbb{E}[Y^2])^{3/2}} = \frac{\mathbb{E}[Y^3]}{\sqrt{\lambda v}\, \mathbb{E}[Y^2]^{3/2}} = \frac{1'030'200'000}{\sqrt{1'000}(1'010'000)^{3/2}} \approx 0.0321.$$

Let F_S denote the distribution function of S. Then, since F_S is continuous and strictly increasing, the quantiles $q_{0.95}$ and $q_{0.99}$ can be calculated as

$$q_{0.95} = F_S^{-1}(0.95) \qquad \text{and} \qquad q_{0.99} = F_S^{-1}(0.99).$$

(a) According to Section 4.1.1 of the lecture notes, the normal approximation is given by

$$F_S(x) \approx \Phi\left(\frac{x - \lambda v \mathbb{E}[Y]}{\sqrt{\lambda v \mathbb{E}[Y^2]}}\right),$$

for all $x \in \mathbb{R}$, where Φ is the standard Gaussian distribution function. For all $\alpha \in (0,1)$ we then have

$$\begin{aligned} F_S^{-1}(\alpha) &= \lambda v \mathbb{E}[Y] + \sqrt{\lambda v \mathbb{E}[Y^2]} \cdot \Phi^{-1}(\alpha) \\ &= 1'000 \cdot 1'000 + \sqrt{1'000 \cdot 1'010'000} \cdot \Phi^{-1}(\alpha) \\ &\approx 1'000'000 + 31'780.5 \cdot \Phi^{-1}(\alpha). \end{aligned}$$

In particular, we get

$$q_{0.95} = F_S^{-1}(0.95) \approx 1'000'000 + 31'780.5 \cdot \Phi^{-1}(0.95) \approx 1'000'000 + 31'780.5 \cdot 1.645 = 1'052'279$$

and

$$q_{0.99} = F_S^{-1}(0.99) \approx 1'000'000 + 31'780.5 \cdot \Phi^{-1}(0.99) \approx 1'000'000 + 31'780.5 \cdot 2.325 = 1'073'890.$$

Note that the normal approximation also allows for negative claims S, which under our model assumption is excluded. The probability for negative claims S in the normal approximation can be calculated as

$$F_S(0) \approx \Phi\left(\frac{0 - \lambda v \mathbb{E}[Y]}{\sqrt{\lambda v \mathbb{E}[Y^2]}}\right) \approx \Phi\left(-\frac{1'000'000}{31'780.5}\right) \approx \Phi(-31.5) \approx 4.34 \cdot 10^{-218},$$

which of course is positive, but very close to 0.

(b) According to Section 4.1.2 of the lecture notes, in the translated gamma approximation we model S by the random variable

$$X = k + Z,$$

where $k \in \mathbb{R}$ and $Z \sim \Gamma(\widetilde{\gamma}, \widetilde{c})$. The three parameters $k, \widetilde{\gamma}$ and $\widetilde{c}$ can be determined by solving the equations

$$\mathbb{E}[X] = \mathbb{E}[S], \qquad \operatorname{Var}(X) = \operatorname{Var}(S) \qquad \text{and} \qquad \varsigma_X = \varsigma_S, \tag{1}$$

where ς_X is the skewness parameter of X. Since $Z \sim \Gamma(\widetilde{\gamma}, \widetilde{c})$, we can use the results given in Section 3.2.1 of the lecture notes to calculate

$$\mathbb{E}[X] = \mathbb{E}[k + Z] = k + \mathbb{E}[Z] = k + \frac{\widetilde{\gamma}}{\widetilde{c}},$$

$$\operatorname{Var}(X) = \operatorname{Var}(k + Z) = \operatorname{Var}(Z) = \frac{\widetilde{\gamma}}{\widetilde{c}^2} \quad \text{and}$$

$$\varsigma_X = \frac{\mathbb{E}\left[(X - \mathbb{E}[X])^3\right]}{\operatorname{Var}(X)^{3/2}} = \frac{\mathbb{E}\left[(k + Z - \mathbb{E}[k + Z])^3\right]}{\operatorname{Var}(k + Z)^{3/2}} = \frac{\mathbb{E}\left[(Z - \mathbb{E}[Z])^3\right]}{\operatorname{Var}(Z)^{3/2}} = \varsigma_Z = \frac{2}{\sqrt{\widetilde{\gamma}}}.$$

Using the equations given in (1), we get

$$\begin{aligned}
\frac{2}{\sqrt{\widetilde{\gamma}}} = \varsigma_S \quad &\iff \quad \widetilde{\gamma} = \frac{4}{\varsigma_S^2} \approx 3'883, \\
\frac{\widetilde{\gamma}}{\widetilde{c}^2} = \operatorname{Var}(S) \quad &\iff \quad \widetilde{c} = \sqrt{\frac{\widetilde{\gamma}}{\operatorname{Var}(S)}} \approx 0.002 \quad \text{and} \\
k + \frac{\widetilde{\gamma}}{\widetilde{c}} = \mathbb{E}[S] \quad &\iff \quad k = \mathbb{E}[S] - \frac{\widetilde{\gamma}}{\widetilde{c}} = \mathbb{E}[S] - \sqrt{\widetilde{\gamma}\operatorname{Var}(S)} \approx -980'392.
\end{aligned}$$

If we write F_Z for the distribution function of $Z \sim \Gamma(\widetilde{\gamma} \approx 3'883, \widetilde{c} \approx 0.002)$, we get using the translated gamma approximation

$$F_S(x) = \mathbb{P}[S \leq x] \approx \mathbb{P}[X \leq x] = \mathbb{P}[k + Z \leq x] = \mathbb{P}[Z \leq x - k] = F_Z(x - k),$$

for all $x \in \mathbb{R}$. Now, for all $\alpha \in (0, 1)$, we have

$$F_S^{-1}(\alpha) \approx k + F_Z^{-1}(\alpha)$$

In particular, we get

$$q_{0.95} = F_S^{-1}(0.95) \approx k + F_Z^{-1}(0.95) \approx -980'392 + 2'032'955 = 1'052'563$$

and

$$q_{0.99} = F_S^{-1}(0.99) \approx k + F_Z^{-1}(0.99) \approx -980'392 + 2'055'074 = 1'074'682.$$

Note that since $k < 0$, the translated gamma approximation in this example also allows for negative claims S, which under our model assumption is excluded. The probability for negative claims S can be calculated as

$$F_S(0) \approx F_Z(0 - k) \approx F_Z(980'392) \approx 4.87 \cdot 10^{-320},$$

which is basically 0.

(c) According to Section 4.1.2 of the lecture notes, in the translated log-normal approximation we model S by the random variable

$$X = k + Z,$$

where $k \in \mathbb{R}$ and $Z \sim \mathrm{LN}(\mu, \sigma^2)$. Similarly as in part (b), the three parameters k, μ and σ^2 can be determined by solving the equations

$$\mathbb{E}[X] = \mathbb{E}[S], \qquad \operatorname{Var}(X) = \operatorname{Var}(S) \qquad \text{and} \qquad \varsigma_X = \varsigma_S. \tag{2}$$

Since $Z \sim \mathrm{LN}(\mu, \sigma^2)$, we can use the results given in Section 3.2.3 of the lecture notes to calculate

$$\begin{aligned}
\mathbb{E}[X] &= \mathbb{E}[k + Z] = k + \mathbb{E}[Z] = k + \exp\left\{\mu + \sigma^2/2\right\}, \\
\operatorname{Var}(X) &= \operatorname{Var}(k + Z) = \operatorname{Var}(Z) = \exp\left\{2\mu + \sigma^2\right\}\left(\exp\left\{\sigma^2\right\} - 1\right) \quad \text{and} \\
\varsigma_X &= \varsigma_Z = \left(\exp\left\{\sigma^2\right\} + 2\right)\left(\exp\left\{\sigma^2\right\} - 1\right)^{1/2}.
\end{aligned}$$

Using the third equation in (2), we get

$$\left(\exp\left\{\sigma^2\right\} + 2\right)\left(\exp\left\{\sigma^2\right\} - 1\right)^{1/2} = \varsigma_S \approx 0.0321 \quad \iff \quad \sigma^2 \approx 0.00011444,$$

which was found using a computer software. Using the second equation in (2), we get

$$\exp\left\{2\mu+\sigma^2\right\}\left(\exp\left\{\sigma^2\right\}-1\right)=\mathrm{Var}(S) \iff \mu=\frac{1}{2}\left(\log\left[\left(\exp\left\{\sigma^2\right\}-1\right)^{-1}\mathrm{Var}(S)\right]-\sigma^2\right),$$

which implies

$$\mu \approx 14.90425.$$

Finally, using the first equation in (2), we get

$$k+\exp\left\{\mu+\sigma^2/2\right\}=\mathbb{E}[S] \quad\iff\quad k=\mathbb{E}[S]-\exp\left\{\mu+\sigma^2/2\right\}\approx -1\text{'}970\text{'}704.$$

If we write F_W for the distribution function of

$$W=\log Z \sim \mathcal{N}(\mu\approx 14.90425, \sigma^2\approx 0.00011444),$$

we get using the translated log-normal approximation

$$F_S(x)=\mathbb{P}[S\le x]\approx\mathbb{P}[X\le x]=\mathbb{P}[k+Z\le x]=\mathbb{P}[\log Z\le\log(x-k)]=F_W(\log[x-k]),$$

for all $x\in\mathbb{R}$. For all $\alpha\in(0,1)$ we then have

$$F_S^{-1}(\alpha)\approx k+\exp\left\{F_W^{-1}(\alpha)\right\}.$$

In particular, we get

$$q_{0.95}=F_S^{-1}(0.95)\approx k+\exp\left\{F_W^{-1}(0.95)\right\}\approx -1\text{'}970\text{'}704+3\text{'}023\text{'}266=1\text{'}052\text{'}562$$

and

$$q_{0.99}=F_S^{-1}(0.99)\approx k+\exp\left\{F_W^{-1}(0.99)\right\}\approx -1\text{'}970\text{'}704+3\text{'}045\text{'}387=1\text{'}074\text{'}684.$$

Note that since $k<0$, the translated log-normal approximation in this example also allows for negative claims S, which under our model assumption is excluded. The probability for negative claims S can be calculated as

$$F_S(0)\approx F_Z(0-k)=F_W(\log[-k])\approx F_W(\log 1\text{'}970\text{'}704)\approx 4.44\cdot 10^{-322},$$

which is basically 0.

(d) We observe that with all the three approximations applied in parts (a) - (c) we get almost the same results. In particular, the normal approximation does not provide estimates that deviate significantly from the ones we get using the translated gamma and the translated log-normal approximations. This is due to the fact that $\lambda v=1\text{'}000$ is large enough and the gamma distribution assumed for the claim sizes is not a heavy tailed distribution. Moreover, the skewness $\varsigma_S=0.0321$ of S is rather small, hence the normal approximation is a valid model in this example. Note that in all the three approximations we allow for negative claims S, which actually should not be possible under our model assumptions. However, the probability to observe a negative claim S is vanishingly small, in all the three approximations.

Solution 7.3 Monte Carlo Simulations

An example of a possible R code for both parts (a) and (b) is given below.

(a) We assume that for this comparably simple problem where no heavy tails are involved 100'000 Monte Carlo simulations are enough to provide an empirical distribution function of S which is close to the true distribution function of S. In Figure 2 we compare the empirical distribution function of S resulting from 100'000 Monte Carlo simulations to the approximate distribution functions when using the normal (left), the translated gamma (middle) and the translated log-normal (right) approximation. From these plots we cannot spot any differences between the various distribution functions.

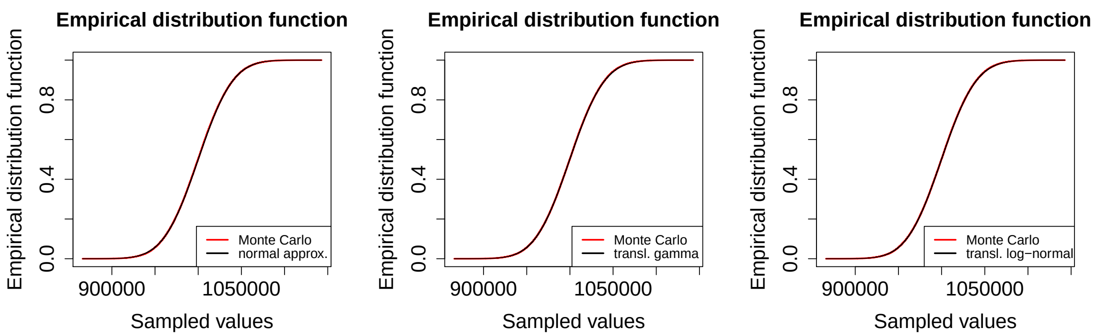

Figure 2: Comparison of the empirical distribution function of S resulting from 100'000 Monte Carlo simulations to the approximate distribution functions when using the normal (left), the translated gamma (middle) and the translated log-normal (right) approximation.

In Figure 3 we consider the log-log plot of the 100'000 Monte Carlo simulations of S and compare it to the normal (left), the translated gamma (middle) and the translated log-normal (right) approximation. We observe that all three approximations have a rather good fit to the tail of the distribution of S, but the translated gamma and the translated log-normal approximation seem slightly more accurate than the normal approximation. We conclude that in the absence of heavy tailed distributions the translated gamma and the translated log-normal approximation are very convincing in this example. Moreover, the skewness of S is small enough ($\varsigma_S \approx 0.0321$, see above) and the expected number of claims large enough ($\lambda v = 1'000$, see Exercise 7.2) for the normal approximation to be a valid approximation, too.

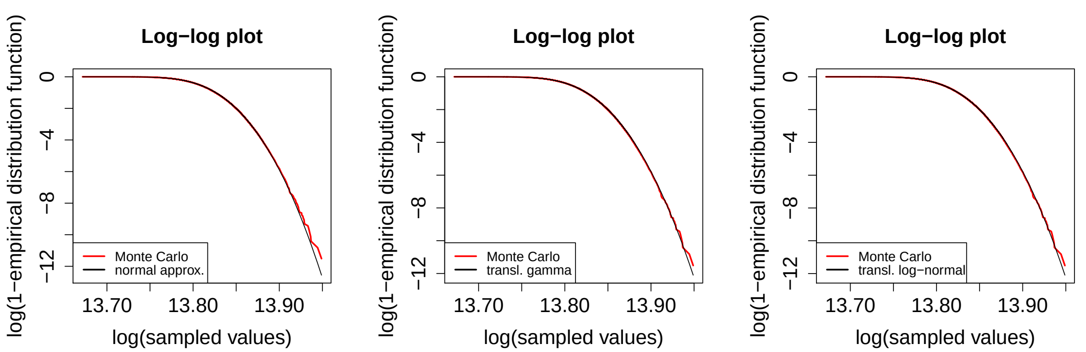

Figure 3: Log-log plot of the 100'000 Monte Carlo simulations of S compared to the normal (left), the translated gamma (middle) and the translated log-normal (right) approximation.

(b) Replicating 10'000 Monte Carlo simulations 100 times already requires some time. This is also the reason why we chose 10'000 as maximum number of simulations and not 100'000 as in part (a). Note that every single time we use Monte Carlo simulations to derive quantities like for example the quantiles $q_{0.95}$ and $q_{0.99}$, we get different results. This is something one needs to be aware of, and it is in contrast to the normal, the translated gamma and the translated log-normal approximation. In Figure 4 we show the densities of the 100 quantiles $q_{0.95}$ (left) and $q_{0.99}$ (right) resulting from replicating the $n = 100, 1'000, 10'000$ Monte Carlo simulations 100 times. We see that increasing the number of simulations n for every replication, the uncertainty regarding the quantiles $q_{0.95}$ and $q_{0.99}$ is reduced.

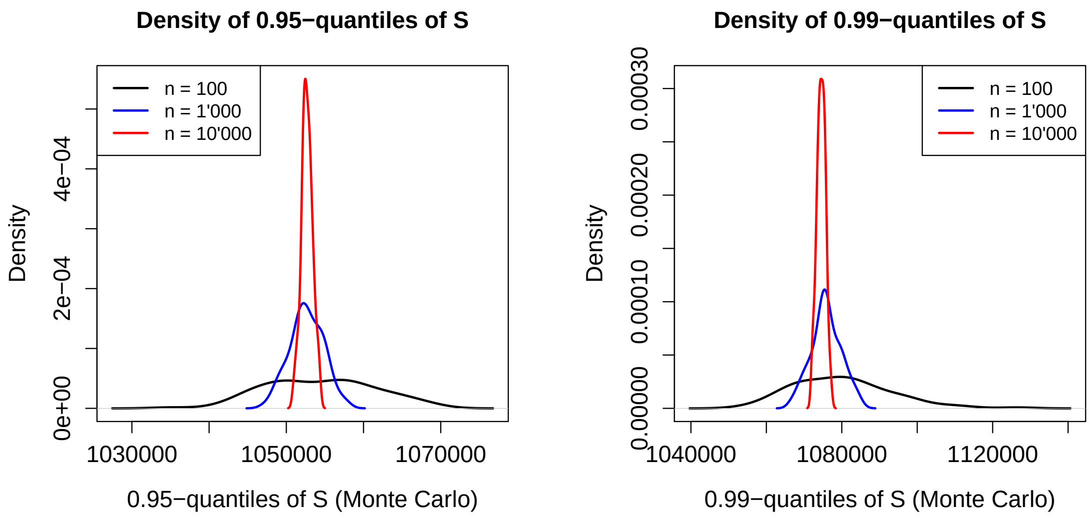

Figure 4: Densities of the 100 quantiles $q_{0.95}$ (left) and $q_{0.99}$ (right) resulting from replicating the $n = 100, 1'000, 10'000$ Monte Carlo simulations 100 times.

	$q_{0.95}$		$q_{0.99}$	
Monte Carlo	smallest	largest	smallest	largest
$n = 100$	1'035'018	1'069'209	1'053'719	1'126'533
$n = 1'000$	1'047'186	1'057'829	1'066'770	1'084'902
$n = 10'000$	1'050'955	1'054'282	1'072'045	1'077'195
Approximations				
normal	1'052'279		1'073'890	
translated gamma	1'052'563		1'074'682	
translated log-normal	1'052'562		1'074'684	

Table 1: Smallest and largest observed values of the quantiles $q_{0.95}$ and $q_{0.99}$ among the 100 replications of the $n = 100, 1'000, 10'000$ Monte Carlo simulations together with the values of the quantiles $q_{0.95}$ and $q_{0.99}$ resulting from the normal, the translated gamma and the translated log-normal approximation.

One can reach the same conclusions from Table 1, where we give the smallest and the largest observed values of the quantiles $q_{0.95}$ and $q_{0.99}$ among the 100 replications of the $n = 100, 1'000, 10'000$ Monte Carlo simulations. Moreover, we also give the values of the quantiles $q_{0.95}$ and $q_{0.99}$ resulting from the normal, the translated gamma and the translated log-normal approximation, see Exercise 7.2. We see that the quantiles resulting from the approximations are always between the smallest and the largest observed value resulting from the Monte Carlo simulations. Of course, one can argue that we could choose the number of simulations n large enough such that the results do not vary considerably anymore. However, a too high number of simulations n will lead to an excessive computation time. This is especially true if one considers heavy tailed distributions. Therefore, one is often inclined to use other algorithms for compound distributions, such as the Panjer algorithm and fast Fourier transforms.

```r
### Function that creates Monte Carlo simulations for the total claim amount S
compound.poisson.distribution <- Vectorize(function(n, lambdav, shape, rate){
  number.of.claims <- rpois(n = n, lambda = lambdav)
  sum(rgamma(n = number.of.claims, shape = shape, rate = rate))
},"n")


### a)

### n Monte Carlo simulations of the total claim amount S
n <- 100000
lambdav <- 1000
shape <- 100
rate <- 1/10
set.seed(100)
claim.amounts <- compound.poisson.distribution(n = rep(1,n), lambdav = lambdav, shape = shape, rate = rate)


### Normal approximation
mu <- lambdav*shape/rate
sigma <- sqrt(lambdav*shape*(shape+1)/(rate^2))

### Check the normal approximation
par(mar=c(5.1, 4.4, 4.1, 2.1))
plot(claim.amounts[order(claim.amounts)], 1:n/(n+1), xlim=c(min(claim.amounts),max(claim.amounts)), type="l", col="red", main="Empirical distribution function", xlab="Sampled values", ylab="Empirical distribution function", cex.lab=1.5, cex.main=1.5, cex.axis=1.5, lwd=2)
lines(claim.amounts[order(claim.amounts)], pnorm((claim.amounts[order(claim.amounts)]),mu,sigma), lwd=1)
legend("bottomright", lty=1, lwd=2, col=c("red","black"), legend=c("Monte Carlo","normal approx.   "), cex=1)
plot(log(claim.amounts[order(claim.amounts)]), log(1-1:n/(n+1)), xlim=c(min(log(claim.amounts)),max(log(claim.amounts))), ylim=c(min(log(1-n/(n+1)),log(1-pnorm((claim.amounts[order(claim.amounts)]),mu,sigma))),0), type="l", col="red", main="Log-log plot", xlab="log(sampled values)", ylab="log(1-empirical distribution function)", cex.lab=1.5, cex.main=1.5, cex.axis=1.5, lwd=2)
lines(log(claim.amounts[order(claim.amounts)]), log(1-pnorm((claim.amounts[order(claim.amounts)]),mu,sigma)), col="black", lwd=1)
```

```
legend("bottomleft", lty=1, lwd=2, col=c("red","black"), legend=c("Monte Carlo","normal approx.   "), cex=1)


### Translated gamma approximation
skews <- (lambdav*shape*(shape+1)*(shape+2)/rate^3)/(lambdav*shape*(shape+1)/rate^2)^(3/2)
shape2 <- 4/skews^2
rate2 <- sqrt(shape2/(lambdav*shape*(shape+1)/rate^2))
k <- lambdav*shape/rate-shape2/rate2

### Check of the translated gamma approximation
plot(claim.amounts[order(claim.amounts)], 1:n/(n+1), xlim=c(min(claim.amounts),max(claim.amounts)), type="l", col="red", main="Empirical distribution function", xlab="Sampled values", ylab="Empirical distribution function", cex.lab=1.5, cex.main=1.5, cex.axis=1.5, lwd=2)
lines(claim.amounts[order(claim.amounts)], pgamma((claim.amounts[order(claim.amounts)])-k,shape=shape2,rate=rate2), lwd=1)
legend("bottomright", lty=1, lwd=2, col=c("red","black"), legend=c("Monte Carlo","transl. gamma   "), cex=1)
plot(log(claim.amounts[order(claim.amounts)]), log(1-1:n/(n+1)), xlim=c(min(log(claim.amounts)),max(log(claim.amounts))), ylim=c(min(log(1-n/(n+1)),log(1-pgamma((claim.amounts[order(claim.amounts)])-k,shape=shape2,rate=rate2))),0), type="l", col="red", main="Log-log plot", xlab="log(sampled values)", ylab="log(1-empirical distribution function)", cex.lab=1.5, cex.main=1.5, cex.axis=1.5, lwd=2)
lines(log(claim.amounts[order(claim.amounts)]), log(1-pgamma((claim.amounts[order(claim.amounts)])-k,shape=shape2,rate=rate2)), lwd=1)
legend("bottomleft", lty=1, lwd=2, col=c("red","black"), legend=c("Monte Carlo","transl. gamma   "), cex=1)


### Translated log-normal approximation
sigma.squared <- 0.00011444
mu2 <- 1/2*(log((exp(sigma.squared)-1)^(-1)*lambdav*shape*(shape+1)/rate^2)-sigma.squared)
k2 <- lambdav*shape/rate-exp(mu2+sigma.squared/2)

### Check of the translated log-normal approximation
plot(claim.amounts[order(claim.amounts)], 1:n/(n+1), xlim=c(min(claim.amounts),max(claim.amounts)), type="l", col="red", main="Empirical distribution function", xlab="Sampled values", ylab="Empirical distribution function", cex.lab=1.5, cex.main=1.5, cex.axis=1.5, lwd=2)
lines(claim.amounts[order(claim.amounts)], pnorm(log((claim.amounts[order(claim.amounts)])-k2),mu2,sqrt(sigma.squared)), lwd=1)
legend("bottomright", lty=1, lwd=2, col=c("red","black"),
```

```r
    legend=c("Monte Carlo","transl. log-normal    "), cex=1)
plot(log(claim.amounts[order(claim.amounts)]), log(1-1:n/(n+1))
    , xlim=c(min(log(claim.amounts)),max(log(claim.amounts))),
    ylim=c(min(log(1-n/(n+1)),log(1-pnorm(log((claim.amounts[
    order(claim.amounts)])-k2),mu2,sqrt(sigma.squared)))),0),
    type="l", col="red", main="Log-log plot", xlab="log(sampled
    values)", ylab="log(1-empirical distribution function)", cex
    .lab=1.5, cex.main=1.5, cex.axis=1.5, lwd=2)
lines(log(claim.amounts[order(claim.amounts)]), log(1-pnorm(log
    ((claim.amounts[order(claim.amounts)])-k2),mu2,sqrt(sigma.
    squared))), lwd=1)
legend("bottomleft", lty=1, lwd=2, col=c("red","black"), legend
    =c("Monte Carlo","transl. log-normal    "), cex=1)



### b)

### k replications of n Monte Carlo Simulations of S
k <- 100
n <- 100
set.seed(100)
claim.amounts.1 <- array(compound.poisson.distribution(n = rep
    (1,k*n), lambdav = 1000, shape = 100, rate = 1/10), dim = c(
    n,k))
n <- 1000
set.seed(200)
claim.amounts.2 <- array(compound.poisson.distribution(n = rep
    (1,k*n), lambdav = 1000, shape = 100, rate = 1/10), dim = c(
    n,k))
n <- 10000
set.seed(300)
claim.amounts.3 <- array(compound.poisson.distribution(n = rep
    (1,k*n), lambdav = 1000, shape = 100, rate = 1/10), dim = c(
    n,k))

### Function that calculates the alpha-quantiles of S on the
    basis of k replications of n Monte Carlo simulations of S
quantiles.monte.carlo <- function(claim.amounts, alpha){
  n <- nrow(claim.amounts)
  claim.amounts.sorted <- apply(claim.amounts, 2, sort)
  quantiles.alpha <- claim.amounts.sorted[floor(alpha*n)+1,]
}

### 0.95-quantiles
quantiles.1 <- quantiles.monte.carlo(claim.amounts = claim.
    amounts.1, alpha = 0.95)
quantiles.2 <- quantiles.monte.carlo(claim.amounts = claim.
    amounts.2, alpha = 0.95)
quantiles.3 <- quantiles.monte.carlo(claim.amounts = claim.
    amounts.3, alpha = 0.95)

```

```
### Minimum/maximum values observed
min(quantiles.1)
max(quantiles.1)
min(quantiles.2)
max(quantiles.2)
min(quantiles.3)
max(quantiles.3)

### Density
ymax <- max(density(quantiles.1)$y, density(quantiles.2)$y,
    density(quantiles.3)$y)
plot(density(quantiles.1), col = "black", ylim = c(0,ymax),
    main = "Density of 0.95-quantiles of S", xlab = "0.95-
    quantiles of S (Monte Carlo)", cex.lab=1.25, cex.main=1.25,
    cex.axis=1.25, lwd=2)
lines(density(quantiles.2), col = "blue", lwd = 2)
lines(density(quantiles.3), col = "red", lwd = 2)
legend("topleft", col=c("black", "blue", "red"), lwd = 2, lty =
     1, legend=c("n = 100","n = 1'000","n = 10'000"))


### 0.99-quantiles
quantiles.1 <- quantiles.monte.carlo(claim.amounts = claim.
    amounts.1, alpha = 0.99)
quantiles.2 <- quantiles.monte.carlo(claim.amounts = claim.
    amounts.2, alpha = 0.99)
quantiles.3 <- quantiles.monte.carlo(claim.amounts = claim.
    amounts.3, alpha = 0.99)

### Minimum/maximum values observed
min(quantiles.1)
max(quantiles.1)
min(quantiles.2)
max(quantiles.2)
min(quantiles.3)
max(quantiles.3)

### Density
ymax <- max(density(quantiles.1)$y, density(quantiles.2)$y,
    density(quantiles.3)$y)
plot(density(quantiles.1), col = "black", ylim = c(0,ymax),
    main = "Density of 0.99-quantiles of S", xlab = "0.99-
    quantiles of S (Monte Carlo)", cex.lab=1.25, cex.main=1.25,
    cex.axis=1.25, lwd=2)
lines(density(quantiles.2), col = "blue", lwd = 2)
lines(density(quantiles.3), col = "red", lwd = 2)
legend("topright", col=c("black", "blue", "red"), lwd = 2, lty
    = 1, legend=c("n = 100","n = 1'000","n = 10'000"))
```

Solution 7.4 Akaike Information Criterion and Bayesian Information Criterion

(a) By definition, the MLEs $\left(\widehat{\gamma}^{\mathrm{MLE}}, \widehat{c}^{\mathrm{MLE}}\right)$ maximize the log-likelihood function $\ell_{\mathbf{Y}}$. In particular, we have

$$\ell_{\mathbf{Y}}\left(\widehat{\gamma}^{\mathrm{MLE}}, \widehat{c}^{\mathrm{MLE}}\right) \geq \ell_{\mathbf{Y}}(\gamma, c),$$

for all $(\gamma, c) \in \mathbb{R}_+ \times \mathbb{R}_+$.

If we write $d^{(\mathrm{MM})}$ and $d^{(\mathrm{MLE})}$ for the number of estimated parameters in the method of moments model and in the MLE model, respectively, we have $d^{(\mathrm{MM})} = d^{(\mathrm{MLE})} = 2$. The AIC value $\mathrm{AIC}^{(\mathrm{MM})}$ of the method of moments model and the AIC value $\mathrm{AIC}^{(\mathrm{MLE})}$ of the MLE model are then given by

$$\mathrm{AIC}^{(\mathrm{MM})} = -2\ell_{\mathbf{Y}}\left(\widehat{\gamma}^{\mathrm{MM}}, \widehat{c}^{\mathrm{MM}}\right) + 2d^{(\mathrm{MM})} = -2 \cdot 1\text{'}264.013 + 2 \cdot 2 = -2\text{'}524.026 \quad \text{and}$$
$$\mathrm{AIC}^{(\mathrm{MLE})} = -2\ell_{\mathbf{Y}}\left(\widehat{\gamma}^{\mathrm{MLE}}, \widehat{c}^{\mathrm{MLE}}\right) + 2d^{(\mathrm{MLE})} = -2 \cdot 1\text{'}264.171 + 2 \cdot 2 = -2\text{'}524.342.$$

According to the AIC, the model with the smallest AIC value should be preferred. Since $\mathrm{AIC}^{(\mathrm{MM})} > \mathrm{AIC}^{(\mathrm{MLE})}$, we choose the MLE fit.

(b) If we write $d^{(\mathrm{gam})}$ and $d^{(\exp)}$ for the number of estimated parameters in the gamma model and in the exponential model, respectively, we have $d^{(\mathrm{gam})} = 2$ and $d^{(\exp)} = 1$. The AIC value $\mathrm{AIC}^{(\mathrm{gam})}$ of the gamma model and the AIC value $\mathrm{AIC}^{(\exp)}$ of the exponential model are then given by

$$\mathrm{AIC}^{(\mathrm{gam})} = -2\ell_{\mathbf{Y}}^{(\mathrm{gam})}\left(\widehat{\gamma}^{\mathrm{MLE}}, \widehat{c}^{\mathrm{MLE}}\right) + 2d^{(\mathrm{gam})} = -2 \cdot 1\text{'}264.171 + 2 \cdot 2 = -2\text{'}524.342 \quad \text{and}$$
$$\mathrm{AIC}^{(\exp)} = -2\ell_{\mathbf{Y}}^{(\exp)}\left(\widehat{c}^{\mathrm{MLE}}\right) + 2d^{(\exp)} = -2 \cdot 1\text{'}264.169 + 2 \cdot 1 = -2\text{'}526.338.$$

Since $\mathrm{AIC}^{(\mathrm{gam})} > \mathrm{AIC}^{(\exp)}$, we choose the exponential model.

The BIC value $\mathrm{BIC}^{(\mathrm{gam})}$ of the gamma model and the BIC value $\mathrm{BIC}^{(\exp)}$ of the exponential model are given by

$$\begin{aligned}\mathrm{BIC}^{(\mathrm{gam})} &= -2\ell_{\mathbf{Y}}^{(\mathrm{gam})}\left(\widehat{\gamma}^{\mathrm{MLE}}, \widehat{c}^{\mathrm{MLE}}\right) + d^{(\mathrm{gam})} \cdot \log 1\text{'}000 \\ &= -2 \cdot 1\text{'}264.171 + 2 \cdot \log 1\text{'}000 \\ &\approx -2\text{'}514.53\end{aligned}$$

and

$$\begin{aligned}\mathrm{BIC}^{(\exp)} &= -2\ell_{\mathbf{Y}}^{(\exp)}\left(\widehat{c}^{\mathrm{MLE}}\right) + d^{(\exp)} \cdot \log 1\text{'}000 \\ &= -2 \cdot 1\text{'}264.169 + \log 1\text{'}000 \\ &\approx -2\text{'}521.43.\end{aligned}$$

According to the BIC, the model with the smallest BIC value should be preferred. Since $\mathrm{BIC}^{(\mathrm{gam})} > \mathrm{BIC}^{(\exp)}$, we choose the exponential model. Note that the gamma model gives the better in-sample fit than the exponential model. But if we adjust this in-sample fit by the number of parameters used, we conclude that the exponential model probably has the better out-of-sample performance (better predictive power).