

# Non-Life Insurance: Mathematics and Statistics

## Solution sheet 8

### Solution 8.1 Panjer Algorithm

For the expected yearly claim amount  $\pi_0$  we have

$$\pi_0 = \mathbb{E}[S] = \mathbb{E}[N] \mathbb{E}[Y_1] = 1 \cdot \mathbb{E}[k + Z] = k + \mathbb{E}[Z] = k + \exp\left\{\mu + \frac{\sigma^2}{2}\right\} \approx 4123.872.$$

Let  $Y_i^+$  denote the discretized claim sizes using a span of  $s = 10$ , where we put all the probability mass to the upper end of the intervals. Note that  $k = 10s$ . If we write  $g_l = \mathbb{P}[Y_1^+ = sl]$  for all  $l \in \mathbb{N}$ , then we have

$$g_1 = g_2 = \dots = g_{10} = 0,$$

since  $\mathbb{P}[Y_1^+ \leq 10s] = \mathbb{P}[k + Z \leq 10s] = \mathbb{P}[Z \leq 0] = 0$ . For all  $l \geq 11$  we get

$$\begin{aligned} g_l &= \mathbb{P}[Y_1^+ = sl] \\ &= \mathbb{P}[Y_1^+ = k + s(l - 10)] \\ &= \mathbb{P}[k + s(l - 11) < Y_1 \leq k + s(l - 10)] \\ &= \mathbb{P}[Y_1 \leq k + s(l - 10)] - \mathbb{P}[Y_1 \leq k + s(l - 11)] \\ &= \mathbb{P}[Z \leq s(l - 10)] - \mathbb{P}[Z \leq s(l - 11)] \\ &= \mathbb{P}[\log Z \leq \log(s[l - 10])] - \mathbb{P}[\log Z \leq \log(s[l - 11])] \\ &= \Phi\left(\frac{\log[s(l - 10)] - \mu}{\sigma}\right) - \Phi\left(\frac{\log[s(l - 11)] - \mu}{\sigma}\right), \end{aligned}$$

where  $\Phi$  is the distribution function of the standard Gaussian distribution and where we define  $\log 0 = -\infty$ . From now on we replace the claim sizes  $Y_i$  with the discretized claim sizes  $Y_i^+$ . By a slight abuse of notation, we still write  $S$  for the yearly claim amount that changed to

$$S = \sum_{i=1}^N Y_i^+.$$

Note that  $N \sim \text{Poi}(1)$  has a Panjer distribution with parameters  $a = 0$  and  $b = 1$ , see Corollary 4.8 of the lecture notes. Applying the Panjer algorithm given in Theorem 4.9 of the lecture notes, we have for  $r \in \mathbb{N}_0$

$$f_r \stackrel{\text{def.}}{=} \mathbb{P}[S = sr] = \begin{cases} \mathbb{P}[N = 0], & \text{for } r = 0, \\ \sum_{l=1}^r \frac{l}{r} g_l f_{r-l}, & \text{for } r > 0. \end{cases}$$

Since the yearly amount that the client has to pay by himself is given by

$$S_{\text{ins}} = \min\{S, d\} + \min\{\alpha \cdot (S - d)_+, M\} = \min\{S, d\} + \alpha \cdot \min\left\{(S - d)_+, \frac{M}{\alpha}\right\},$$

$M/\alpha = 7'000$  and the maximal possible franchise is  $2'500$ , we have to apply the Panjer algorithm until we reach  $\mathbb{P}[S = 9'500] = f_{950}$ . Here we limit ourselves to determine the values of  $f_0, \dots, f_{12}$  to illustrate how the algorithm works. We have

$$f_0 = \mathbb{P}[N = 0] = e^{-1} \approx 0.37$$

and

$$f_1 = f_2 = \dots = f_{10} = 0,$$

since  $g_1 = g_2 = \dots = g_{10} = 0$ . For  $r = 11$  and  $r = 12$  we get

$$f_{11} = \sum_{l=1}^{11} \frac{l}{11} g_l f_{11-l} = g_{11} f_0 = \left[ \Phi \left( \frac{\log s - \mu}{\sigma} \right) - \Phi \left( \frac{\log 0 - \mu}{\sigma} \right) \right] e^{-1} \approx 7.089 \cdot 10^{-9}$$

and

$$f_{12} = \sum_{l=1}^{12} \frac{l}{12} g_l f_{12-l} = g_{12} f_0 = \left[ \Phi \left( \frac{\log 2s - \mu}{\sigma} \right) - \Phi \left( \frac{\log s - \mu}{\sigma} \right) \right] e^{-1} \approx 2.786 \cdot 10^{-7}.$$

Using the discretized claim sizes, the yearly expected amount  $\pi_{\text{ins}}$  paid by the customer is given by

$$\pi_{\text{ins}} = \mathbb{E}[S_{\text{ins}}] = \mathbb{E}[\min\{S, d\}] + \alpha \mathbb{E} \left[ \min \left\{ (S - d)_+, \frac{M}{\alpha} \right\} \right],$$

where we have

$$\mathbb{E}[\min\{S, d\}] = \sum_{r=0}^{d/s} f_r sr + d \left( 1 - \sum_{r=0}^{d/s} f_r \right) = d + \sum_{r=0}^{d/s} f_r (sr - d)$$

and

$$\begin{aligned} \mathbb{E} \left[ \min \left\{ (S - d)_+, \frac{M}{\alpha} \right\} \right] &= \sum_{r=d/s+1}^{d/s+M/s\alpha} f_r (sr - d) + \frac{M}{\alpha} \left( 1 - \sum_{r=0}^{d/s+M/s\alpha} f_r \right) \\ &= \frac{M}{\alpha} + \sum_{r=d/s+1}^{d/s+M/s\alpha} f_r \left( sr - d - \frac{M}{\alpha} \right) - \frac{M}{\alpha} \sum_{r=0}^{d/s} f_r. \end{aligned}$$

Therefore, we get

$$\begin{aligned} \pi_{\text{ins}} &= d + \sum_{r=0}^{d/s} f_r (sr - d) + \alpha \left[ \frac{M}{\alpha} + \sum_{r=d/s+1}^{d/s+M/s\alpha} f_r \left( sr - d - \frac{M}{\alpha} \right) - \frac{M}{\alpha} \sum_{r=0}^{d/s} f_r \right] \\ &= d + M + \sum_{r=0}^{d/s} f_r (sr - d - M) + \sum_{r=d/s+1}^{d/s+M/s\alpha} \alpha f_r \left( sr - d - \frac{M}{\alpha} \right). \end{aligned}$$

Finally, if the customer has chosen franchise  $d$ , then the monthly pure risk premium  $\pi$  is given by

$$\begin{aligned} \pi &= \frac{\pi_0 - \pi_{\text{ins}}}{12} \\ &= \frac{1}{12} \left[ k + \exp \left\{ \mu + \frac{\sigma^2}{2} \right\} - d - M - \sum_{r=0}^{d/s} f_r (sr - d - M) - \sum_{r=d/s+1}^{d/s+M/s\alpha} \alpha f_r \left( sr - d - \frac{M}{\alpha} \right) \right]. \end{aligned}$$

In the end, we get the following monthly pure risk premiums  $\pi$  for the different franchises  $d$ :

franchise $d$	300	500	1'000	1'500	2'000	2'500
monthly pure risk premium $\pi$	307	297	274	253	233	216

Table 1: Monthly pure risk premiums  $\pi$  for the different franchises  $d$ .

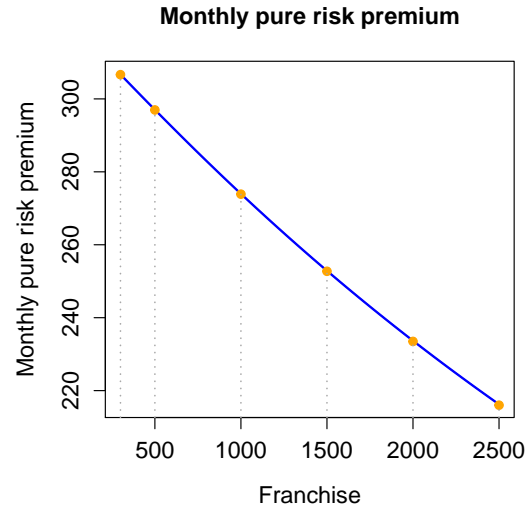


Figure 1: Plot of the monthly pure risk premium  $\pi$  as a function of the franchise  $d$ .

More generally, the monthly pure risk premium  $\pi$  as a function of the franchise  $d$ , which is allowed to vary between 300 CHF and 2'500 CHF, is given in Figure 1. Note that these monthly premiums only represent the pure risk premiums. In order get the premiums that the customer has to pay in the end, we would need to add an appropriate risk-loading, which may vary between different health insurance companies. The above plot can be created by the R code given below, where we calculated the premiums using two different discretizations of the claim sizes: in one we put the probability mass to the upper end of the intervals and in the other to the lower end of the intervals. However, the resulting premiums for these two versions are basically the same.

```

1  ### Define the function KK_premium with the variables:
2  ### lambda = mean number of claims
3  ### mu = mean parameter of log-normal distribution
4  ### sigma2 = variance parameter of log-normal distribution
5  ### span = span size used in the Panjer algorithm
6  ### shift = shift of the translated log-normal distribution
7  KK_premium <- function(lambda, mu, sigma2, span, shift){
8
9     ### We will calculate the distribution of S until M (M = 2500 +
10    7000)
11    M <- 9500
12
13    ### Number of steps
14    m <- floor(M/span)
15
16    ### We won't have any mass until we reach shift, which happens at
17    the k0-th step
18    k0 <- shift/span
19
20    ### Initialize array where mass is put to the lower end of the
21    interval
22    g_min <- array(0, dim=c(m+1,1))
    
```

```

20
21   ### Initialize array where mass is put to the upper end of the
      interval
22   g_max <- array(0, dim=c(m+1,1))
23
24   ### Discretize the log-normal distribution putting the mass to
      the lower end of the interval
25   for (k in (k0+1):(m+1)){g_min[k,1] <- pnorm(log((k-k0)*span),
      mean=mu, sd=sqrt(sigma2))-pnorm(log((k-k0-1)*span), mean=mu,
      sd=sqrt(sigma2))}
26
27   ### Discretize the log-normal distribution putting the mass to
      the upper end of the interval
28   g_max[2:(m+1),1] <- g_min[1:m,1]
29
30   ### Initialize matrix, where we will store the probability
      distribution of S
31   f1 <- matrix(0, nrow=m+1, ncol=3)
32
33   ### Store the probability of getting zero claims (in both lower
      bound and upper bound)
34   f1[1,1] <- exp(-lambda*(1-g_min[1,1]))
35   f1[1,2] <- exp(-lambda*(1-g_max[1,1]))
36
37   ### Calculate the values "l * g_{l}" of the discretized claim
      sizes (lower bound and upper bound), we need these values in
      the Panjer algorithm
38   h1 <- matrix(0, nrow=m, ncol=3)
39   for (i in 1:m){
40     h1[i,1] <- g_min[i+1,1]*(i+1)
41     h1[i,2] <- g_max[i+1,1]*(i+1)
42   }
43
44   ### Panjer algorithm (note that in the Poisson case we have a = 0
      and b = lambda*v, which is just lambda here)
45   for (r in 1:m){
46     f1[r+1,1] <- lambda/r*(t(f1[1:r,1])%*%h1[r:1,1])
47     f1[r+1,2] <- lambda/r*(t(f1[1:r,2])%*%h1[r:1,2])
48     f1[r+1,3] <- r * span
49   }
50
51   ### Maximal and minimal franchise
52   m1 <- 2500
53   m0 <- 300
54
55   ### Number of iterations needed to get to m1 and m0
56   i1 <- floor(m1/span+1)
57   i0 <- floor(m0/span+1)
58
59   ### Calculate the part that the insured pays by himself
60   franchise <- array(NA, c(i1, 3))
61   for (i in i0:i1){

```

```

62 franchise[i,1] <- f1[i,3]    ### this represents the franchise
63 franchise[i,2] <- sum(f1[1:i,1]*f1[1:i,3]) + f1[i,3] * (1-sum(
    f1[1:i,1]))
64 franchise[i,2] <- franchise[i,2] + sum(f1[(i+1):floor(i+7000/
    span),1]*f1[2:floor(7000/span+1),3])*0.1 + 700 * (1-sum(f1
    [1:floor(i+7000/span),1]))
65 franchise[i,3] <- sum(f1[1:i,2]*f1[1:i,3]) + f1[i,3] * (1-sum(
    f1[1:i,2]))
66 franchise[i,3] <- franchise[i,3] + sum(f1[(i+1):floor(i+7000/
    span),2]*f1[2:floor(7000/span+1),3])*0.1 + 700 * (1-sum(f1
    [1:floor(i+7000/span),2]))
67 }
68
69 ### Calculate the price of the monthly premium
70 price <- array(NA, c(i1, 3))
71 price[,1] <- franchise[,1]    ### this represents the franchise
72 price[,2:3] <- (lambda*(exp(mu+sigma2/2)+shift) - franchise
    [,2:3])/12
73 price
74 }
75
76 ### Load the add-on packages stats and MASS
77 require(stats)
78 require(MASS)
79
80 ### Determine values for the input parameters of the function KK_
    premium
81 lambda <- 1
82 mu <- 7.8
83 sigma2 <- 1
84 span <- 10
85 shift <- 100
86
87 ### The coefficient of variation of the translated log-normal
    distribution is given by
88 exp(mu+sigma2/2)*sqrt(exp(sigma2)-1)/(shift+exp(mu+sigma2/2))
89
90 ### Run the function KK_premium
91 price <- KK_premium(lambda, mu, sigma2, span, shift)
92
93 ### Plot the monthly pure risk premium as a function of the
    franchise
94 plot(x=price[,1], y=price[,2], lwd=2, col="blue", type='l', ylab="
    Monthly pure risk premium", xlab="Franchise", main="Monthly pure
    risk premium", cex.lab=1.25, cex.main=1.25, cex.axis=1.25)
95 lines(x=price[,1], y=price[,2], lwd=1, col="blue")
96 points(x=c(300,500, 1000, 1500, 2000, 2500), y=price[c(300,500,
    1000, 1500, 2000, 2500)/span+1,3], pch=19, col="orange")
97 lines(x=c(300,300), y=c(0,price[300/span+1,3]),lty=3, lwd=1.5, col=
    "darkgray")
98 lines(x=c(500,500), y=c(0,price[500/span+1,3]),lty=3, lwd=1.5, col=
    "darkgray")

```

```

99 lines(x=c(1000,1000), y=c(0,price[1000/span+1,3]),lty=3, lwd=1.5,
      col="darkgray")
100 lines(x=c(1500,1500), y=c(0,price[1500/span+1,3]),lty=3, lwd=1.5,
      col="darkgray")
101 lines(x=c(2000,2000), y=c(0,price[2000/span+1,3]),lty=3, lwd=1.5,
      col="darkgray")
102 lines(x=c(2500,2500), y=c(0,price[2500/span+1,3]),lty=3, lwd=1.5,
      col="darkgray")
103
104 ### Give the monthly pure risk premiums for the six franchises
      listed on the exercise sheet
105 round(price[floor(c(300, 500, 1000, 1500, 2000, 2500)/span+1),2])
106 round(price[floor(c(300, 500, 1000, 1500, 2000, 2500)/span+1),3])
    
```

### Solution 8.2 Fast Fourier Transform

Assume that  $\tilde{Y}$  follows the claim size distribution given on the exercise sheet. Let  $Y$  denote the discretized version of  $\tilde{Y}$  that takes values in  $\mathbb{N}_0$ . More precisely, we shift the probability masses of  $\tilde{Y}$  to the right and define

$$\mathbb{P}[Y = 0] = 0 \quad \text{and} \quad \mathbb{P}[Y = l] = \mathbb{P}[\tilde{Y} \leq l] - \mathbb{P}[\tilde{Y} \leq l - 1],$$

for all  $l \in \mathbb{N}$ . By a slight abuse of notation, we still write  $S$  for the compound Poisson distribution with discrete claim sizes distributed as  $Y$ . In particular, also  $S$  takes values in  $\mathbb{N}_0$ . We define

$$g_l = \mathbb{P}[Y = l] \quad \text{and} \quad f_l = \mathbb{P}[S = l],$$

for all  $l \in \mathbb{N}_0$ . We choose a threshold of  $n = 2'000'000$ , i.e. we determine the distribution function of  $S$  up to  $n - 1$ . Note that  $n$  is chosen sufficiently high such that we approximately have

$$P[Y > n - 1] \approx 0. \tag{1}$$

We define  $\mathcal{A} = \{0, \dots, n - 1\}$  and calculate the discrete Fourier transform  $(\hat{g}_z)_{z \in \mathcal{A}}$  of  $(g_l)_{l \in \mathcal{A}}$  by

$$\hat{g}_z = \sum_{l=0}^{n-1} g_l \exp \left\{ 2\pi i \frac{zl}{n} \right\}, \tag{2}$$

for all  $z \in \mathcal{A}$ . Due to (1), we approximately have

$$\hat{g}_z \approx \mathbb{E} \left[ \exp \left\{ 2\pi i \frac{zY}{n} \right\} \right] = M_Y \left( 2\pi i \frac{z}{n} \right),$$

for all  $z \in \mathcal{A}$ , where  $M_Y$  denotes the moment generating function of  $Y$ . Note that we use an extended version of the moment generating function also allowing for complex numbers. If  $M_S$  denotes the moment generating function of  $S$ , then, according to Proposition 2.11 of the lecture notes (again extended to complex numbers), we have

$$M_S \left( 2\pi i \frac{z}{n} \right) = \exp \left\{ \lambda v \left[ M_Y \left( 2\pi i \frac{z}{n} \right) - 1 \right] \right\} \approx \exp \{ \lambda v (\hat{g}_z - 1) \}, \tag{3}$$

for all  $z \in \mathcal{A}$ . The left hand side of equation (3) can be written as

$$M_S \left( 2\pi i \frac{z}{n} \right) = \sum_{l=0}^{\infty} f_l \exp \left\{ 2\pi i \frac{zl}{n} \right\} = \sum_{l=0}^{n-1} \left( f_l + \sum_{k=1}^{\infty} f_{l+kn} \right) \exp \left\{ 2\pi i \frac{zl}{n} \right\},$$

for all  $z \in \mathcal{A}$ . Using the approximation

$$f_l \approx f_l + \sum_{k=1}^{\infty} f_{l+kn}, \quad (4)$$

for all  $l \in \mathcal{A}$ , we compute the discrete Fourier transform  $(\hat{f}_z)_{z \in \mathcal{A}}$  of  $(f_l)_{l \in \mathcal{A}}$  by

$$\hat{f}_z = \sum_{l=0}^{n-1} f_l \exp \left\{ 2\pi i \frac{zl}{n} \right\} \approx M_S \left( 2\pi i \frac{z}{n} \right) \approx \exp \{ \lambda v (\hat{g}_z - 1) \},$$

for all  $z \in \mathcal{A}$ . Applying the inversion formula of the discrete Fourier transform, we finally calculate

$$f_l = \frac{1}{n} \sum_{z=0}^{n-1} \hat{f}_z \exp \left\{ -2\pi i \frac{zl}{n} \right\}, \quad (5)$$

for all  $l \in \mathcal{A}$ . Note that due to the approximation in (4), instead of  $f_l$  we actually calculate

$$f_l + \sum_{k=1}^{\infty} f_{l+kn} > f_l,$$

for all  $l \in \mathcal{A}$ . This error is called wrap around error (or aliasing error), and  $n$  should be chosen large enough in order to keep this wrap around error small. In R, the calculations in equations (2) and (5) can be done using the command `fft`. An example of a possible R code is given below. In Figure (2) we compare the distribution function (left) and the log-log plot (right) of  $S$  resulting from the fast Fourier transform algorithm to the Monte Carlo simulations. We see that we get a very good fit. In particular, the threshold  $n = 2'000'000$  seems to be high enough.

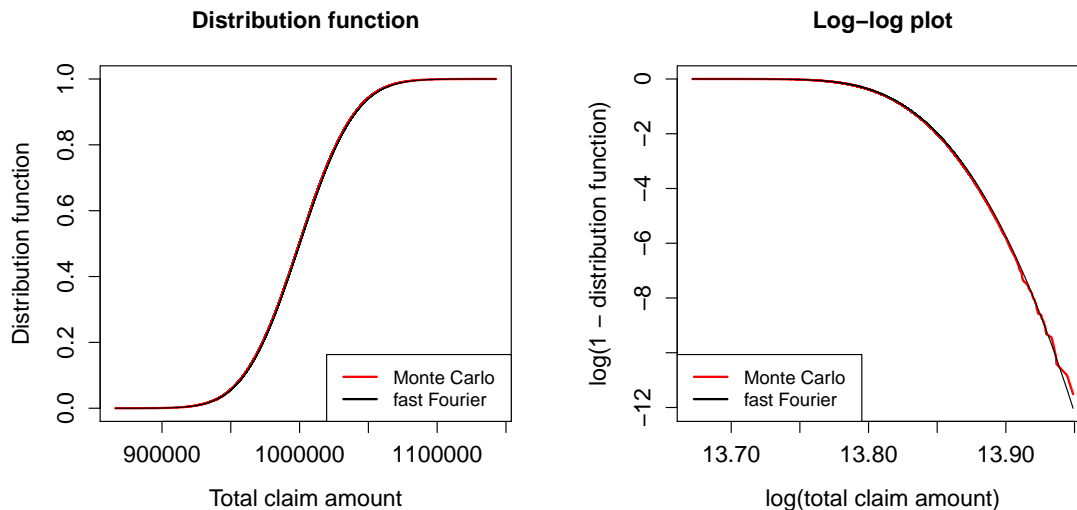


Figure 2: Comparison of the distribution function (left) and the log-log plot (right) of  $S$  resulting from the fast Fourier transform algorithm to the Monte Carlo simulations.

For the 0.95-quantile  $q_{0.95}$  and the 0.99-quantile  $q_{0.99}$  we get

$$q_{0.95} = 1'053'089 \quad \text{and} \quad q_{0.99} = 1'075'215.$$

We see that we get values which are very close to the ones derived in Exercise 7.2, where we used the normal, the translated gamma and the translated log-normal approximation.

```

1  ### Fast Fourier transform
2  n <- 2000000
3  lambdav <- 1000
4  prob.claim.size <- c(0, pgamma(1:(n-1), shape=100, rate=1/10)-
   pgamma(0:(n-2), shape=100, rate=1/10))
5  prob.claim.size.ft <- fft(prob.claim.size)
6  prob.total.claim.amount.ft <- exp(lambdav*(prob.claim.size.ft-1))
7  prob.total.claim.amount <- Re(fft(prob.total.claim.amount.ft,
   inverse=TRUE)/length(prob.total.claim.amount.ft))
8
9  ### a)
10
11 ### Monte Carlo simulations from Exercise 7.3
12 compound.poisson.distribution <- Vectorize(function(n, lambdav,
   shape, rate){
13   number.of.claims <- rpois(n = n, lambda = lambdav)
14   sum(rgamma(n = number.of.claims, shape = shape, rate = rate))
15 }, "n")
16 m <- 100000
17 set.seed(100)
18 claim.amounts <- compound.poisson.distribution(n = rep(1,m),
   lambdav = 1000, shape = 100, rate = 1/10)
19
20 ### Calculate the values of the distribution function of S using
   the fast Fourier transform results
21 probabilities <- cumsum(prob.total.claim.amount)[floor(claim.
   amounts[order(claim.amounts)])+1]
22
23 ### Check the fast Fourier transform result
24 par(mar=c(5.1, 4.4, 4.1, 2.1))
25 plot(claim.amounts[order(claim.amounts)], 1:m/(m+1), xlim=c(min(
   claim.amounts),max(claim.amounts)), type="l", col="red", main="
   Distribution function", xlab="Total claim amount", ylab="
   Distribution function", cex.lab=1.25, cex.main=1.25, cex.axis
   =1.25, lwd=2)
26 lines(claim.amounts[order(claim.amounts)], probabilities, lwd=1)
27 legend("bottomright", lty=1, lwd=2, col=c("red","black"), legend=c(
   "Monte Carlo","fast Fourier  "), cex=1)
28 plot(log(claim.amounts[order(claim.amounts)]), log(1-1:m/(m+1)),
   xlim=c(min(log(claim.amounts)),max(log(claim.amounts))), ylim=c(
   min(log(1-m/(m+1)),log(1-probabilities)),0), type="l", col="red"
   , main="Log-log plot", xlab="log(total claim amount)", ylab="log
   (1 - distribution function)", cex.lab=1.25, cex.main=1.25, cex.
   axis=1.25, lwd=2)
29 lines(log(claim.amounts[order(claim.amounts)]), log(1-probabilities
   ), col="black", lwd=1)
30 legend("bottomleft", lty=1, lwd=2, col=c("red","black"), legend=c("
   Monte Carlo","fast Fourier  "), cex=1)
31
32 ### b)
33
34 ### Determine the 0.95- and the 0.99-quantiles

```



```

35 which(cumsum(prob.total.claim.amount) > 0.95) [1] -1
36 which(cumsum(prob.total.claim.amount) > 0.99) [1] -1
    
```

### Solution 8.3 Variance Loading Principle

- (a) Let  $S_1, S_2, S_3$  be the total claim amounts of the passenger cars, delivery vans and trucks, respectively. Then, according to Proposition 2.11 of the lecture notes, we have

$$\mathbb{E}[S_i] = \lambda_i v_i \mathbb{E} \left[ Y_1^{(i)} \right],$$

for all  $i \in \{1, 2, 3\}$ . Using the data given in the table on the exercise sheet, we get

$$\begin{aligned} \mathbb{E}[S_1] &= 0.25 \cdot 40 \cdot 2'000 = 20'000, \\ \mathbb{E}[S_2] &= 0.23 \cdot 30 \cdot 1'700 = 11'730 \quad \text{and} \\ \mathbb{E}[S_3] &= 0.19 \cdot 10 \cdot 4'000 = 7'600. \end{aligned}$$

If we write  $S$  for the total claim amount of the car fleet, we can conclude that

$$\mathbb{E}[S] = \mathbb{E}[S_1 + S_2 + S_3] = \mathbb{E}[S_1] + \mathbb{E}[S_2] + \mathbb{E}[S_3] = 39'330.$$

- (b) Again using Proposition 2.11 of the lectures notes, we get

$$\begin{aligned} \text{Var}[S_i] &= \lambda_i v_i \mathbb{E} \left[ \left( Y_1^{(i)} \right)^2 \right] = \lambda_i v_i \left( \text{Var} \left( Y_1^{(i)} \right) + \mathbb{E} \left[ Y_1^{(i)} \right]^2 \right) \\ &= \lambda_i v_i \mathbb{E} \left[ Y_1^{(i)} \right]^2 \left( \text{Vco}(Y_1^{(i)})^2 + 1 \right), \end{aligned}$$

for all  $i \in \{1, 2, 3\}$ . Using the data given in the table on the exercise sheet, we find

$$\begin{aligned} \text{Var}(S_1) &= 0.25 \cdot 40 \cdot 2'000^2 \cdot (2.5^2 + 1) = 290'000'000, \\ \text{Var}(S_2) &= 0.23 \cdot 30 \cdot 1'700^2 \cdot (2^2 + 1) = 99'705'000 \quad \text{and} \\ \text{Var}(S_3) &= 0.19 \cdot 10 \cdot 4'000^2 \cdot (3^2 + 1) = 304'000'000. \end{aligned}$$

Since  $S_1, S_2$  and  $S_3$  are independent by assumption, we get for the variance of the total claim amount  $S$  of the car fleet

$$\text{Var}(S) = \text{Var}(S_1) + \text{Var}(S_2) + \text{Var}(S_3) = 693'705'000.$$

Using the variance loading principle with  $\alpha = 3 \cdot 10^{-6}$ , we get for the premium  $\pi$  of the car fleet

$$\pi = \mathbb{E}[S] + \alpha \text{Var}(S) = 39'330 + 3 \cdot 10^{-6} \cdot 693'705'000 \approx 39'330 + 2'081 = 41'411.$$

Note that we have

$$\frac{\pi - \mathbb{E}[S]}{\mathbb{E}[S]} = \frac{\alpha \text{Var}(S)}{\mathbb{E}[S]} \approx \frac{2'081}{39'330} \approx 5.3\%.$$

Thus, the loading  $\pi - \mathbb{E}[S]$  is given by 5.3% of the pure risk premium.

**Solution 8.4 Panjer Distribution**

If we write

$$p_k = \mathbb{P}[N = k],$$

for all  $k \in \mathbb{N}$ , then, by definition of the Panjer distribution, we have

$$p_k = p_{k-1} \left( a + \frac{b}{k} \right),$$

for all  $k \in \mathbb{N}$ . We can use this recursion to calculate  $\mathbb{E}[N]$  and  $\text{Var}(N)$ . Note that the range of  $N$  is  $\mathbb{N}$  if  $a \geq 0$  and it is  $\{0, 1, \dots, n\}$  for some  $n \in \mathbb{N}_{\geq 1}$  if  $a < 0$ .

First, we consider the case where  $a < 0$ , i.e. where the range of  $N$  is  $\{0, 1, \dots, n\}$ . According to the proof of Lemma 4.7 of the lecture notes, we have

$$n = -\frac{a+b}{a}. \tag{6}$$

For the expectation of  $N$  we get

$$\begin{aligned} \mathbb{E}[N] &= \sum_{k=0}^n k p_k = \sum_{k=1}^n k p_k = \sum_{k=1}^n k p_{k-1} \left( a + \frac{b}{k} \right) = a \sum_{k=1}^n k p_{k-1} + b \sum_{k=1}^n p_{k-1} \\ &= a \sum_{k=0}^{n-1} (k+1) p_k + b \sum_{k=0}^{n-1} p_k = a \sum_{k=0}^{n-1} k p_k + (a+b) \sum_{k=0}^{n-1} p_k \\ &= a (\mathbb{E}[N] - n p_n) + (a+b)(1 - p_n) \\ &= a \mathbb{E}[N] + a + b + p_n(-an - a - b). \end{aligned}$$

Using (6), we get

$$-an - a - b = a \frac{a+b}{a} - a - b = 0. \tag{7}$$

Hence, the above expression for  $\mathbb{E}[N]$  simplifies to

$$\mathbb{E}[N] = a\mathbb{E}[N] + a + b,$$

from which we can conclude that

$$\mathbb{E}[N] = \frac{a+b}{1-a}.$$

In order to get the variance of  $N$ , we first calculate the second moment of  $N$ :

$$\begin{aligned} \mathbb{E}[N^2] &= \sum_{k=0}^n k^2 p_k = \sum_{k=1}^n k^2 p_k = \sum_{k=1}^n k^2 p_{k-1} \left( a + \frac{b}{k} \right) \\ &= a \sum_{k=1}^n k^2 p_{k-1} + b \sum_{k=1}^n k p_{k-1} = a \sum_{k=0}^{n-1} (k+1)^2 p_k + b \sum_{k=0}^{n-1} (k+1) p_k \\ &= a \sum_{k=0}^{n-1} k^2 p_k + (2a+b) \sum_{k=0}^{n-1} k p_k + (a+b) \sum_{k=0}^{n-1} p_k \\ &= a (\mathbb{E}[N^2] - n^2 p_n) + (2a+b)(\mathbb{E}[N] - n p_n) + (a+b)(1 - p_n) \\ &= a \mathbb{E}[N^2] + (2a+b) \mathbb{E}[N] + a + b + p_n[-an^2 - (2a+b)n - a - b]. \end{aligned}$$

Using (6), we get

$$\begin{aligned}
 -an^2 - (2a + b)n - a - b &= -a \left( \frac{a + b}{a} \right)^2 + (2a + b) \frac{a + b}{a} - a - b \\
 &= -\frac{a^2 + 2ab + b^2}{a} + \frac{2a^2 + 3ab + b^2}{a} - \frac{a^2 + ab}{a} \\
 &= 0.
 \end{aligned} \tag{8}$$

Hence, the above expression for  $\mathbb{E}[N^2]$  simplifies to

$$\mathbb{E}[N^2] = a \mathbb{E}[N^2] + (2a + b) \mathbb{E}[N] + a + b,$$

from which we get

$$\begin{aligned}
 \mathbb{E}[N^2] &= \frac{(2a + b) \mathbb{E}[N] + a + b}{1 - a} \\
 &= \frac{(2a + b)(a + b) + (a + b)(1 - a)}{(1 - a)^2} \\
 &= \frac{2a^2 + 3ab + b^2 + a - a^2 + b - ab}{(1 - a)^2} \\
 &= \frac{(a + b)^2 + a + b}{(1 - a)^2}.
 \end{aligned}$$

Finally, the variance of  $N$  then is

$$\text{Var}(N) = \mathbb{E}[N^2] - \mathbb{E}[N]^2 = \frac{(a + b)^2 + a + b}{(1 - a)^2} - \frac{(a + b)^2}{(1 - a)^2} = \frac{a + b}{(1 - a)^2}.$$

In the case where  $a \geq 0$ , i.e. where the range of  $N$  is  $\mathbb{N}$ , we can perform analogous calculations with the only difference that the index of summation in all the sums involved goes up to  $\infty$  instead of stopping at  $n$ . As a consequence, the calculations in (7) and in (8) aren't necessary anymore. The formulas for  $\mathbb{E}[N]$  and  $\text{Var}(N)$ , however, remain the same.

The ratio of  $\text{Var}(N)$  to  $\mathbb{E}[N]$  is given by

$$\frac{\text{Var}(N)}{\mathbb{E}[N]} = \frac{a + b}{(1 - a)^2} \frac{1 - a}{a + b} = \frac{1}{1 - a}.$$

Note that if  $a < 0$ , i.e. if  $N$  has a binomial distribution, we have  $\text{Var}(N) < \mathbb{E}[N]$ . If  $a = 0$ , i.e. if  $N$  has a Poisson distribution, we have  $\text{Var}(N) = \mathbb{E}[N]$ . Finally, in the case of  $a > 0$ , i.e. for a negative-binomial distribution, we have  $\text{Var}(N) > \mathbb{E}[N]$ .