UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Mark Trobina

Posplošeni Panjerjev algoritem in numerična stabilnost pri izračunu tveganj

Delo diplomskega seminarja

Mentor: doc. dr. Marjetka Krajnc
Somentor: asist. dr. Gregor Šega

Ljubljana, 2014

Kazalo

Posplošeni Panjerjev algoritem in numerična stabilnost pri izračunu tveganj

Povzetek

V uvodu je predstavljen problem tveganja in negotovosti, s katerim se soočajo vse zavarovalnice. Vpeljan je matematični model, pri katerem sta definirani slučajni spremenljivki, ki predstavljata število in višino škodnih zahtevkov, obrazloženih pa je tudi nekaj ključnih predpostavk. Cilj je ugotoviti porazdelitev skupnega zneska, ki ga mora pokriti zavarovalnica. Na kratko sta razložena dva pristopa k rešitvi, ki pa nista optimalna. Kot rešitev predstavim Panjerjev algoritem in navedem slučajne spremenljivke, ki pridejo v upoštev za uporabo v modelu in za katere pravimo, da spadajo v Panjerjev razred. Pri izvedbi algoritma pa lahko pride do numeričnih težav. Razlogi za to so podrobno razloženi, numerične napake pa so ponazorjene z nekaj zgledi. Predstavljen je tudi posplošeni Panjerjev algoritem, s pomočjo katerega se razvije numerično stabilen algoritem za dve vrsti porazdelitev, ki sta tudi podrobneje obrazloženi. Sledi nekaj besed, ki so namenjene uporabi v praksi. Opisani so tudi primeri porazdelitev, ki se jih najbolj pogosto uporablja. Za zaključek pa je opisan zgled na konkretnih podatkih, kjer je uporabljeno pridobljeno znanje. S pomočjo dobljenih rezultatov lahko bolje ocenimo tveganje in negotovost, kar je bistveno za uspešno poslovanje zavarovalnic in drugih finančnih institucij.

Generalization of Panjer's algorithm and numerically stable risk aggregation

Abstract

The problem of risk and uncertainty, which is inevitable for insurance companies, is first discussed. Mathematical model of this problem is then presented and claim number distribution and claim distribution are defined. Some key assumptions are made. The main goal is to compute the distribution of all claims combined. Two out of three solutions are then shortly described, but they are not optimal. The third solution is the Panjer's algorithm, which is also explained in addition to the applicable random variables. These variables are said to belong to the Panjer class. Numerical instabilities may arise when the Panjer's algorithm is performed and the reasons for that are well explained. Moreover, examples are provided to illustrate numerical difficulties. The modified Panjer's algorithm follows and this solves the numerical problems for two distributions from the Panjer class. These distributions are also explained in detail. Practical applications are discussed and most popular distributions are stated. To conclude, an example based on real data is presented, using the knowledge gained in this paper. With the help of these results, risk can be better managed, which is essential for insurance companies and other financial institutions as well.

Ključne besede: Panjerjev algoritem, aktuarstvo, upravljanje s tveganjem
Keywords: Panjer's algorithm, actuarial science, risk management

1. Uvod

V svetu financ sta negotovost in tveganje ključnega pomena. Z njima se ukvarjajo praktično vse finančne institucije, zato je razumevanje teh dveh pojmov toliko bolj pomembno. Nerazumevanje ali morda celo podcenjevanje tveganj lahko hitro vodi v velike izgube. Finančne institucije imajo seveda za glavni cilj ustvarjanje dobička. Banke zbirajo denar svojih klientov in jim v zameno ponujajo neko nizko obrestno mero. Ta denar potem posojajo podjetjem ali gospodinjstvom in zahtevajo nekoliko višjo obrestno mero. Razlika v obrestnih merah predstavlja njihov profit. Zdi se precej enostavno, toda problem se pojavi, ko podjetje ali gospodinjstvo svojega posojila ne poplača. To je tveganje, ki ga banke morajo znati čim bolje oceniti in nato ustrezno poslovati.

Naslednja finančna institucija, ki je morda za moj diplomski seminar še bolj pomembna, je zavarovalnica. Zavarovanci zavarovalnici plačajo premijo, ta pa jim v zameno zagotavlja, da v primeru škodnega dogodka krije del ali pa celotno škodo, ki je nastala. Število škodnih dogodkov ter posamezne škode pa so negotove in podobno kot banke morajo tudi zavarovalnice znati oceniti to tveganje. Če želijo imeti dobiček, morajo biti premije dovolj visoke, škodnih zahtevkov pa ne sme biti preveč. Tu je potrebno omeniti tudi to, da je za zavarovalnice zelo pomembno, da ima dovolj veliko število zavarovancev, saj je potem zaradi zakonov velikih števil lažje ocenjevati tveganja.

Podobno velja tudi za ostale finančne institucije. Vse imajo obveznosti, ki pa niso vnaprej znane. V nadaljevanju se bom osredotočil na primer zavarovalnic, saj se Panjerjev algoritem večinoma uporablja v zavarovalništvu.

1.1. **Matematični okvir.**

Vse slučajne spremenljivke, ki nastopajo v tem delu, so definirane na nekem fiksnem verjetnostnem prostoru $(\Omega, \mathcal{F}, P)$. Za začetek definirajmo dve vrsti slučajnih spremenljivk:

- N ... število vseh škodnih zahtevkov
- X_i ... višina $i-$tega škodnega zahtevka

Slučajna spremenljivka N je očitno diskretna z zalogo vrednosti v $\mathbb{N}_0$. Slučajna spremenljivka X_i pa je načeloma zvezna, a bomo predpostavili, da je prav tako diskretna z zalogo vrednosti v $\mathbb{N}_0$. To predpostavko bomo kasneje podrobneje utemeljili, potrebujemo pa jo zaradi izreka, ki pride kasneje in zahteva, da je X_i aritmetična slučajna spremenljivka.

Definicija 1.1.
Slučajna spremeljivka X je aritmetična, če za nek $d > 0$ velja

$$P[X \in \{kd; k \in \mathbb{Z}\}] = 1.$$

V našem primeru je ob zgoraj omenjeni predpostavki X_i aritmetična, d je enak 1.

Zavarovalnico najbolj zanima, koliko denarja bo morala plačati za poplačilo vseh škod v nekem časovnem obdobju, torej slučajna vsota

$$S = \sum_{n=1}^{N} X_n.$$

Glede na zgoraj opisano ima tudi S zalogo vrednosti v naravnih številih z dodano ničlo. Predpostaviti je treba tudi dve zelo pomembni stvari, ki zelo olajšata oziroma najbrž kar omogočita izračun verjetnostne funkcije S. Ti predpostavki se glasita:

- $X_1, X_2, \ldots$ so enako porazdeljene
- $N, X_1, X_2, \ldots$ so neodvisne

V nadaljevanju bo veljalo $X_1, X_2, \ldots \sim X$. Da bi zavarovalnica optimalno poslovala, je zelo pomembno, da pozna porazdelitev slučajne vsote S.

2. Izračun porazdelitve slučajne vsote S

2.1. **Monte Carlo simulacije.**

Prvi način, s katerim bi lahko izračunali verjetnost, da je slučajna vsota enaka nekemu celemu številu m, je s simulacijami spremenljivk. Poznamo namreč verjetnostni funkciji N in X. Najprej simuliramo N, nato N-krat simuliramo še X in jih seštejemo ter zabeležimo, koliko znaša ta vsota. Ta postopek ponovimo n-krat in na koncu izračunamo približno verjetnostno funkcijo slučajne vsote S po formuli

$$P[S = m] \doteq \frac{\#\text{ poskusov, kjer } S = m}{n}.$$

Problem tega pristopa je, da dobimo le približke pravih vrednosti. Natančnost se seveda povečuje z večanjem števila ponovitev poskusa, a za 10-krat večjo natančnost potrebujemo kar 100-krat več poskusov. V praksi je zaradi tega željena raven natančnosti enostavno nedosegljiva.

2.2. **Direkten pristop.**

Naslednji način je zelo neposreden, za nadaljevanje pa je treba definirati slučajne spremenljivke

$$S_1 = X_1$$

in

$$S_k = S_{k-1} + X_k \quad \text{za } k \geq 2.$$

Verjetnostne funkcije teh spremenljivk dobimo s konvolucijo na sledeč način:

$$P[S_k = n] = \sum_{i=0}^{n} P[S_{k-1} = i, X_k = n - i] = \sum_{i=0}^{n} P[S_{k-1} = i]P[X_k = n - i],$$

kjer drugi enačaj velja zaradi neodvisnosti $S_{k-1} = X_1 + ... + X_{k-1}$ in X_k.

Sedaj je potrebno razmisliti, kako lahko pride do tega, da je slučajna vsota S enaka n. To se lahko zgodi na več načinov. Lahko imamo k škodnih zahtevkov, kjer se $X_1, ..., X_k$ seštejejo ravno v n. Zaradi disjunktnosti teh dogodkov in neodvisnosti N in S_k sledi

$$P[S = n] = \sum_{k=0}^{\infty} P[N = k]P[S_k = n].$$

Tako smo dobili željeno verjetnostno funkcijo, ki se jo da izračunati numerično stabilno. Toda težava tega pristopa je, da je časovno zelo zahteven, saj je časovna zahtevnost enaka $O(n^3)$. S tem še nismo zadovoljni in treba bo poiskati drugo pot.

3. Panjerjev pristop

Predstavljeni sta bili že dve metodi za izračun porazdelitve S, zdaj pa je čas za glaven algoritem, ki se imenuje po Kanadčanu z imenom Harry Panjer. Panjerjev algoritem je rekurziven postopek, ki izračuna verjetnostno funkcijo slučajne vsote S. Pred natančnejšim opisom algoritma, pa je treba povedati, da je nujno predpostaviti, da slučajna spremenljivka N spada v razred $Panjer(a, b, k)$.

3.1. **Razred Panjer(a,b,k).**

Definicija 3.1.
Slučajna sprem eljivka N spada v razred Panjer(a, b, k), če obstajajo $a, b \in \mathbb{R}$ in $k \in \mathbb{N}_0$, da velja

$$\begin{aligned} &P[N = 0] = ... = P[N = k - 1] = 0, \\ &P[N = k] \neq 0, \\ &P[N = n] = \left(a + \frac{b}{n}\right) P[N = n - 1], \; n > k. \end{aligned}$$

Vidimo, da ta zahteva ni tako trivialna. Poglejmo si nekaj porazdelitev, ki so v tem razredu, in predstavimo njihove parametre a, b, k, saj je to zelo pomembno pri obravnavi numerične stabilnosti.

Trditev 3.2. *Porazdelitve v Panjerjevem razredu so sledeče:*

(a) $Bin(n, p)$ *v razredu* $Panjer(\frac{p}{p-1}, \frac{(n+1)p}{1-p}, 0)$
(b) $NegBin(\alpha, p)$ *v razredu* $Panjer(1 - p, (1 - p)(\alpha - 1), 0)$
(c) $Pois(\lambda)$ *v razredu* $Panjer(0, \lambda, 0)$
(d) $Log(q)$ *v razredu* $Panjer(q, -q, 1)$
(e) $ExtNegBin(\alpha, k, p)$ *v razredu* $Panjer(1 - p, (1 - p)(\alpha - 1), k)$
(f) $ExtLog(k, q)$ *v razredu* $Panjer(q, -kq, k)$

Dokaz.
(a) Očitno velja

$$P[N = 0] = (1 - p)^n \neq 0 \;\; \text{za} \;\; p < 1.$$

Dalje je

$$\begin{aligned} \frac{P[N = k]}{P[N = k - 1]} &= \frac{\binom{n}{k} p^k (1 - p)^{n-k}}{\binom{n}{k-1} p^{k-1} (1 - p)^{n-k+1}} = \\ &= \frac{p(n - k + 1)}{(1 - p)k} = \frac{p}{1 - p} \left(\frac{n + 1}{k} - 1\right) \end{aligned}$$

in to mora biti enako $\left(a+\frac{b}{k}\right)$. Iz tega sledi, da sta

$$a=\frac{p}{p-1}, \quad b=\frac{(n+1)p}{1-p}.$$

(b) Spet je očitno, da velja

$$P[N=0]=p^{\alpha}\neq 0 \quad \text{za } p>0.$$

Velja tudi

$$\frac{P[N=k]}{P[N=k-1]}=\frac{\frac{\Gamma(\alpha+k)}{k!\Gamma(\alpha)}p^{a}(1-p)^{k}}{\frac{\Gamma(\alpha+k-1)}{(k-1)!\Gamma(\alpha)}p^{a}(1-p)^{k-1}}=$$

$$=\frac{\alpha+k-1}{k}(1-p)=(1-p)+\frac{\alpha-1}{k}(1-p),$$

kar mora biti enako $\left(a+\frac{b}{k}\right)$. Sledi

$$a=1-p, \quad b=(1-p)(\alpha-1).$$

(c) Očitno velja

$$P[N=0]=e^{-\lambda}\neq 0.$$

hkrati pa je res tudi

$$\frac{P[N=k]}{P[N=k-1]}=\frac{e^{-\lambda}\frac{\lambda^{k}}{k!}}{e^{-\lambda}\frac{\lambda^{k-1}}{(k-1)!}}=\frac{\lambda}{k}.$$

To mora biti enako $\left(a+\frac{b}{k}\right)$ in to nam da $a=0$ in $b=\lambda$

(d) Seveda drži

$$P[N=1]=\frac{-q}{\ln(1-q)}\neq 0 \quad \text{za } q\in(0,1).$$

Dalje je

$$\frac{P[N=n]}{P[N=n-1]}=\frac{\frac{-1}{\ln(1-q)}\frac{q^{n}}{n}}{\frac{-1}{\ln(1-q)}\frac{q^{n-1}}{n-1}}=\frac{q(n-1)}{n},$$

kar mora biti enako $\left(a+\frac{b}{n}\right)$ in zato

$$a=q, \quad b=-q.$$

(e) Po definiciji velja

$$P[N=j]=0 \text{ za } j=0,\ldots,k-1$$

in

$$P[N=k]=\frac{\binom{\alpha+k-1}{k}(1-p)^{k}}{p^{-\alpha}-\sum_{j=0}^{k-1}\binom{\alpha+j-1}{j}(1-p)^{j}}\neq 0 \text{ za } p\in(0,1).$$

Velja tudi

$$\frac{P[N=n]}{P[N=n-1]} = \frac{\frac{\binom{\alpha+n-1}{n}(1-p)^n}{p^{-\alpha}-\sum_{j=0}^{k-1}\binom{\alpha+j-1}{j}(1-p)^j}}{\frac{\binom{\alpha+n-2}{n-1}(1-p)^{n-1}}{p^{-\alpha}-\sum_{j=0}^{k-1}\binom{\alpha+j-1}{j}(1-p)^j}} = \frac{(1-p)(\alpha+n-1)}{n}$$

in to mora biti enako $\left(a+\frac{b}{n}\right)$. Zato sledi

$$a = 1-p, \quad b = (1-p)(\alpha-1).$$

(f) Po definiciji je

$$P[N=j] = 0 \text{ za } j = 0, \ldots, k-1$$

in

$$P[N=k] = \frac{q^k}{\sum_{\ell=k}^{\infty}\binom{\ell}{k}^{-1}q^\ell} \neq 0 \text{ za } q > 0.$$

Dalje je

$$\frac{P[N=n]}{P[N=n-1]} = \frac{\frac{\binom{n}{k}^{-1}q^n}{\sum_{\ell=k}^{\infty}\binom{\ell}{k}^{-1}q^\ell}}{\frac{\binom{n-1}{k}^{-1}q^{n-1}}{\sum_{\ell=k}^{\infty}\binom{\ell}{k}^{-1}q^\ell}} = q\frac{n-k}{n}.$$

To mora biti enako$\left(a+\frac{b}{n}\right)$ in zato sledi

$$a = q, \quad b = -kq.$$

$\square$

Zgornja trditev nam torej pove, katere slučajne spremenljivke lahko predstavljajo število škodnih zahtevkov, če računamo porazdelitev S po Panjerjevemu algoritmu, ki je opisan spodaj.

3.2. **Osnovni Panjerjev algoritem.**

Trditev 3.3.
Naj bo N v razredu Panjer(a,b,k) in $aP[X=0] \neq 1$. Potem lahko izračunamo porazdelitev slučajne vsote S na naslednji način:

$$P[S=0] = \begin{cases} P[N=0] & ; P[X=0]=0 \\ E[(P[X=0])^N] & ; \text{sicer} \end{cases} \tag{1}$$

$$P[S=n] = \frac{\left(P[S_k=n]P[N=k] + \sum_{j=1}^{n}(a+\frac{bj}{n})P[X=j]P[S=n-j]\right)}{1-aP[X=0]} \tag{2}$$

za vse $n \in \mathbb{N}$.

Dokaz trditve sledi iz dokaza splošnejšega izreka 4.1.

Časovna zahtevnost algoritma je $O(n^2)$, kar je ogromna prednost pred direktnim pristopom, kjer je bila $O(n^3)$.

3.3. **Težave Panjerjevega algoritma.**

Prvi problem, ki se pojavi pri algoritmu, je izračun verjetnosti $P[S=0]$, ki je seveda potrebna za izvedbo rekurzije. Očitno je, da bo ta verjetnost res zelo majhna. Če izberemo npr. $N \sim Pois(\lambda)$, se po algoritmu izračuna

$$P[S=0] = E[P[X_1=0]^N] = \sum_{k=0}^{\infty} P[X_1=0]^k e^{-\lambda}\frac{\lambda^k}{k!} =$$

$$= e^{-\lambda}\sum_{k=0}^{\infty}\frac{(P[X_1=0]\lambda)^k}{k!} = e^{-\lambda(1-P(X_1=0))}.$$

Ko modeliramo velik portfelj zavarovalnih pogodb, je število škodnih zahtevkov običajno veliko in zato moramo vzeti velik λ, kar pa pomeni, da je $P[S=0]$ izredno majhna. Hitro potem pride do tega, da to verjetnost lahko na računalniku predstavimo le kot nič, saj imamo omejen obseg predstavljivih števil in pride do podkoračitve. To pa zaradi rekurzivne zveze pomeni, da je $P[S=n]=0$ za vsak n, kar pa ni res.

Ta problem rešimo tako, da namesto λ uporabimo parameter $\lambda' := \lambda/2^m$, kjer m izberemo tako, da je potem $P[S=0]$ predstavljivo število na računalniku. To deluje, saj vemo, da za neodvisne slučajne spremenljivke

$$N_1, \dots, N_{2^m} \sim Pois(\lambda/2^m)$$

velja

$$N = N_1 + \cdots + N_{2^m} \sim Pois(\lambda).$$

Podobno se znajdemo pri preostalih porazdelitvah N.

Drugi in bolj pomemben problem pa je v členu $(a+\frac{bj}{n})$, ki se pojavlja v (2). Lahko se namreč zgodi, da omenjeni člen spremeni predznak, ko j v vsoti teče od 1 do n. Ker je le ta pomnožen še z dvema verjetnostima, ki sta običajno zelo majhni, pride potem do odštevanja dveh majhnih števil in posledično do izgubljenih točnih decimalnih mest in napačnih rezultatov. Zaradi rekurzivne zveze se nato napake le še kopičijo in tako pride do numerične nestabilnosti. Omenjen problem si poglejmo bolj podrobno.

Algoritem je stabilen, če je

$$a + \frac{bj}{n} \geq 0, \text{ za vse } j \in \{1, \dots, n\}.$$

To se zgodi, ko $a \geq 0$ in $b+a \geq 0$, torej za:

- $NegBin(\alpha, p)$ v razredu $Panjer(1-p, (1-p)(\alpha-1), 0)$

  Prvi pogoj je očiten, drugi pogoj pa velja, saj je

  $$(1-p)(\alpha-1) + (1-p) = \alpha - p\alpha.$$

  Ker je $p \in (0,1)$, je to res večje od nič.

- $Pois(\lambda)$ v razredu $Panjer(0, \lambda, 0)$

  Prvi pogoj je zopet očiten, drugi pa drži, ker je $\lambda \geq 0$.

- $Log(q)$ v razredu $Panjer(q, -q, 1)$

  Oba pogoja sta jasno izpolnjena, saj je $q \geq 0$.

Opomba 3.4.
V primeru negativne binomske in Poissonove porazdelitve velja, da se zaokrožitvena napaka, ki se kopiči tekom poteka algoritma, povečuje le linearno.
Poleg tega velja, da če zaokrožujemo na r decimalnih mest, je število točnih decimalk $\#(r, n)$ za $P[S = n]$ omejeno s stopnjo zaupanja 99% z

$$\#(r, n) \geq r + \left\lfloor \log_{10} \frac{4}{3} - \frac{1}{2} \log_{10}(n + 1) \right\rfloor .$$

Za podrobnosti glej [4].

Algoritem je potencialno nestabilen v primerih, ko se predznak spremeni. To se zgodi za:

- $Bin(n, p)$ v razredu $Panjer(\frac{p}{p-1}, \frac{(n+1)p}{1-p}, 0)$

  Prva neenakost ne drži, saj je $\frac{p}{p-1} < 0$.

- $ExtNegBin(\alpha, k, p)$ v razredu $Panjer(1 - p, (1 - p)(\alpha - 1), k)$

  Druga neenakost ne drži, saj je

  $$(1 - p)(\alpha - 1) + (1 - p) = \alpha - p\alpha$$

  in

  $$\alpha \in (-k, -k + 1),\ \ p > 0,\ \ k \in \mathbb{N}.$$

  Torej je to manjše od nič.

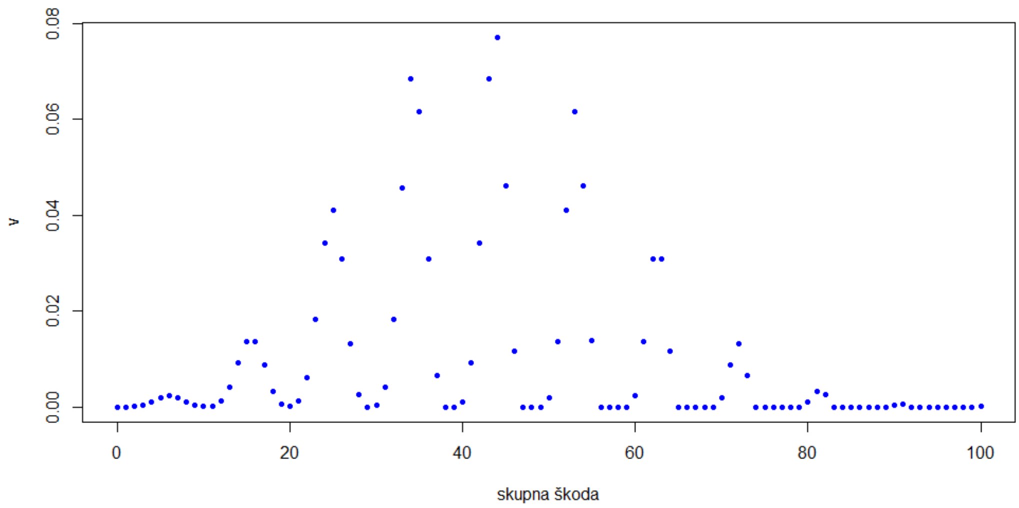
- $ExtLog(k, q)$ v razredu $Panjer(q, -kq, k)$

  Druga neenakost ne drži, saj je $q - kq < 0$.

Za ilustracijo si oglejmo primer, kjer pride do numeričnih napak.

Primer 3.5.
Naj bo $N \sim Bin(10, 0.75)$ in $X \sim \begin{pmatrix} 1 & 10 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$.

Torej imamo le 10 zavarovancev, ki pa z visoko verjetnostjo 0.75 razbijejo avto. Višina škode je ali majhna (1) ali pa visoka (10) z enako verjetnostjo. S Panjerjevim algoritmom izračunamo porazdelitev slučajne vsote S, pri čemer uporabimo aritmetiko z dvojno natančnostjo, kar je približno 10^{-16}. Na naslednji strani je prikazan graf verjetnostne funkcije.



Na prvi pogled izgleda graf verjetnostne funkcije v redu. Vrednosti so med 0 in 100, verjetnosti so nekako smiselno razporejene. V povprečju bo avto razbilo 7.5 strank, povprečna škoda pa bo 5.5. Zmnožek je enak 41.25 in tam okoli so verjetnosti največje tudi na grafu. Toda ni vse v najlepšem redu. Zaradi binomske porazdelitve N-ja pride do odštevanja majhnih vrednosti in do čudnih rezultatov. Verjetnost, da mora zavarovalnica pokriti škodo 38 bi morala biti enaka nič, saj je nemogoče z največ 10 škodnimi zahtevki, visokimi bodisi 1 bodisi 10, dobiti vsoto 38. Ampak v algoritmu pride do odštevanja majhnih števil, ko računamo $P[S = 38]$ in dobimo pozitivno število. To število je sicer zelo blizu 0, toda vpliva na vse ostale člene. Zaradi podobnih težav pride celo do negativnih verjetnosti za nekatere k. Recimo

$$P[S = 99] = -0.0000000146.$$

Vse to posledično privede tudi do tega, da je vsota vseh vrednosti verjetnostne funkcije večja od 1 in sicer 1.00000001128. V tem primeru torej Panjerjev algoritem ne izračuna čisto prave porazdelitve za slučajno vsoto S. Dobljene verjetnosti ter graf so dobljene iz spodnje kode v programu R.

```
primer <- function(n,p){
  verjetnosti <- c((1-p)^n)
  X <- rbind(1:(10*n),rep(0,10*n))
  X[2,c(1,10)] <- c(1/2,1/2)
  a <- -p/(1-p)
  b <- (n+1)*p/(1-p)
  for(i in 1:(10*n)){
    vsota <- 0
    for(j in 1:i){
      vsota <- vsota + (a+b*j/i)*X[2,j]*verjetnosti[i-j+1]
    }
    verjetnosti <- c(verjetnosti,vsota)
  }
  return(verjetnosti)
}
```

Za tri izmed šestih možnih porazdelitev, ki pripadajo razredu $Panjer(a, b, k)$, je torej algoritem iz trditve 3.3 numerično nestabilen. Ogledali si bomo izboljšan algoritem, ki odpravi težave za dve izmed treh porazdelitev. Še prej pa poglejmo, kako je definirana posplošena negativna binomska oziroma posplošena logaritemska porazdelitev.

4.1. **Posplošena negativna binomska porazdelitev.**

Naj bo $N \sim ExtNegBin(a, k, p)$, kjer so $k \in \mathbb{N}, \alpha \in (-k, -k+1)$ in $p \in (0, 1)$. Verjetnostna funkcija je definirana kot:

$$P[N = n] = \begin{cases} \frac{\binom{\alpha+n-1}{n}(1-p)^n}{p^{-\alpha} - \sum_{j=0}^{k-1}\binom{\alpha+j-1}{j}(1-p)^j} & ; n \geq k \\ 0 & ; \text{sicer} \end{cases}$$

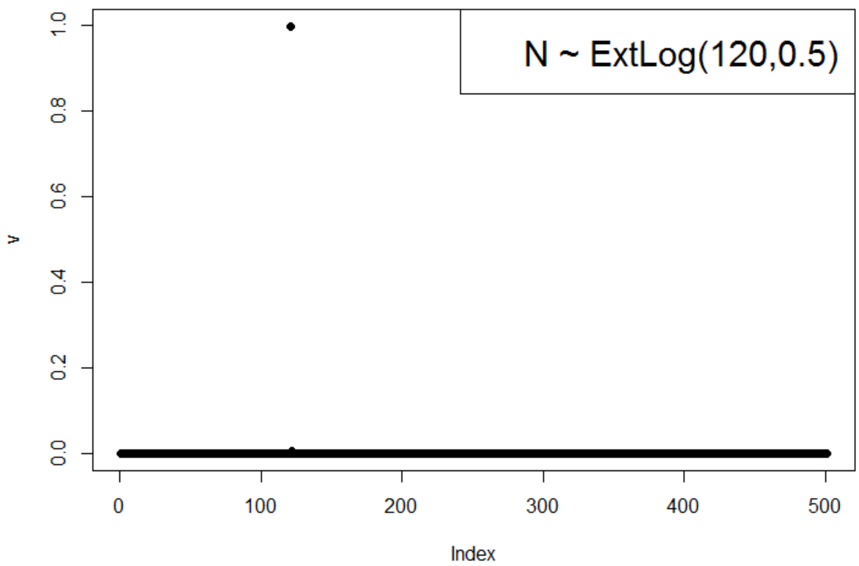
Opazimo tudi, da velja $ExtNegBin(\alpha, 0, p) = NegBin(\alpha, p)$. Bolj zanimivo pa je dejstvo, da je pri tej porazdelitvi večina verjetnosti skoncentrirana pri k. Recimo, da velja $\alpha = -9.9$, $k = 10$ in $p = 0.1$. Potem je $P[N = 10] = 0.9910985$, kar pomeni, da je res zelo malo verjetno, da $N \neq 10$.

4.2. **Posplošena logaritemska porazdelitev.**

Naj bo $N \sim ExtLog(k, q)$, kjer $k \in \mathbb{N} \setminus \{1\}$ in $q \in (0, 1)$. Verjetnostna funkcija je definirana kot:

$$P[N = n] = \begin{cases} \frac{\binom{n}{k}^{-1} q^n}{\sum_{\ell=k}^{\infty} \binom{\ell}{k}^{-1} q^\ell} & ; n \geq k \\ 0 & ; \text{sicer} \end{cases}$$

Opazimo, da je $ExtLog(1, q) = Log(q)$. Podobno kot prej je večina verjetnosti skoncentrirana pri k. Recimo, da sta $k = 120$ in $q = 0.5$.
Potem je $P[N = 120] = 0.9958508$ in $P[N = 121] = 0.004115086$. Spet bo N z veliko verjetnostjo pokazal 120. Spodnji graf to lepo prikazuje.



4.3. **Posplošeni Panjerjev algoritem.**

Želimo si izračunati verjetnostno funkcijo slučajne vsote S tudi za ti dve porazdelitvi in osnova je spodnji izrek, ki je bolj splošen od trditve 3.3.

Izrek 4.1.
Naj bo $\ell \in \mathbb{N}$ in naj bodo slučajne spremenljivke N in $\widetilde{N}_i$ za $i \in \{1, \ldots, \ell\}$, ki predstavljajo število škodnih zahtevkov, neodvisne od zaporedja $\{X_n\}_{n\in\mathbb{N}}$ neodvisnih in enako porazdeljenih ($\sim X$) slučajnih spremeljivk, ki predstavljajo višino škodnih zahtevkov.
Definirajmo

$$S = X_1 + \ldots + X_N$$

in

$$\widetilde{S}_i = X_1 + \ldots + X_{\widetilde{N}_i}$$

za $i \in \{1, \ldots, \ell\}$.

Recimo, da obstajajo $k \in \mathbb{N}_0$ in $a_1, \ldots a_\ell, b_1, \ldots, b_\ell \in \mathbb{R}$, da velja

$$P[\widetilde{N}_i = 0] = \ldots = P[\widetilde{N}_i = k + \ell - i - 1] = 0 \quad za \quad i \in \{1, \ldots, \min(\ell, k + \ell - 1)\} \tag{3}$$

in velja tudi

$$P[N = n] = \sum_{i=1}^{\ell} \left(a_i + \frac{b_i}{n}\right) P\left[\widetilde{N}_i = n - i\right] \quad za \quad n \geq k + \ell. \tag{4}$$

Potem je

$$P[S = 0] = \begin{cases} P[N = 0] & ; P[X = 0] = 0 \\ E[(P(X = 0])^N] & ; \text{sicer} \end{cases} \tag{5}$$

in

$$P[S = n] = \sum_{j=1}^{k+\ell-1} P[S_j = n]P[N = j] + \sum_{i=1}^{\ell}\sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in}\right) P\left[S_i = j\right] P\left[\widetilde{S}_i = n - j\right] \tag{6}$$

za $n \in \mathbb{N}$, kjer je $S_i = X_1 + \ldots + X_i$.

Dokaz.
Dokažimo najprej enakost (5). Za začetek razbijemo dogodek, da je vsota enaka nič, na dva disjunktna dogodka

$$\{S = 0\} = \{S = 0, N = 0\} \cup \{S = 0, N \geq 1\}.$$

Nato gremo direktno računati verjetnost

$$P[S = 0] = P[S = 0, N = 0] + P[S = 0, N \geq 1].$$

Uporabimo dejstvo, da je

$$\{S = 0, N = 0\} = \{N = 0\},$$

in

$$\{S = 0, N \geq 1\} = \{X_1 = 0, \ldots, X_N = 0, N \geq 1\}.$$

To pomeni, da je

$$P[S=0]=P[N=0]+P[X_1=0,\ldots,X_N=0,N\geq 1].$$

Zdaj ločimo primere.

(a) $P[X=0]=0$

Zaradi neodvisnosti se druga verjetnost razbije na produkt verjetnosti in je enaka nič. Torej je

$$P[S=0]=P[N=0],$$

kar smo želeli pokazati.

(b) $P[X=0]>0$

V tem primeru pa uporabimo neodvisnost X_i in N ter enako porazdeljenost $\{X_i\}_{n\in\mathbb{N}}\sim X$ in zato velja

$$P[S=0]=P[N=0]+\sum_{n=1}^{\infty}P[X=0]^nP[N=n]=$$
$$=\sum_{n=0}^{\infty}P[X=0]^nP[N=n]=E[P[X=0]^N].$$

Enakost (5) je s tem dokazana, zato nadaljujemo z dokazom enakosti (6). Izberemo poljuben $n\in\mathbb{N}$. Nato si izberemo še $i\in\{1,\ldots,\ell\}$ in definiramo za vsak $m\in\mathbb{N},\ m\geq i$ spremenljivko

$$S_m=X_1+\ldots+X_m=S_{m-i}+S_{i,m}\ ,$$

kjer je

$$S_{i,m}=X_{m-i+1}+...+X_m.$$

Zdaj izpeljemo naslednjo enakost, ki bo potrebna v nadaljevanju dokaza:

$$
\begin{aligned}
a_i+\frac{b_i}{m}&=E[a_i+\frac{b_i}{m}\frac{S_m}{n}\mid S_m=n]=\\
&=E[a_i\mid S_m=n]+\frac{b_i}{mn}E\left[S_m\mid S_m=n\right]=\\
&=E[a_i\mid S_m=n]+\frac{b_i}{mn}\sum_{j=1}^{m}E[X_j\mid S_m=n]=\\
&=E[a_i\mid S_m=n]+\frac{b_i}{n}E[X_m\mid S_m=n]=\\
&=E[a_i\mid S_m=n]+\frac{b_i}{in}E[iX_m\mid S_m=n]=\\
&=E[a_i\mid S_m=n]+\frac{b_i}{in}E[X_{m-i+1}+...+X_m\mid S_m=n]=\\
&=E[a_i\mid S_m=n]+\frac{b_i}{in}E[S_{i,m}|\ S_m=n]=\\
&=E\left[a_i+\frac{b_iS_{i,m}}{in}\mid S_m=n\right]=\sum_{j=0}^{n}\left(a_i+\frac{b_ij}{in}\right)P[S_{i,m}=j\mid S_m=n]
\end{aligned}
$$

Torej velja:

$$a_i + \frac{b_i}{m} = \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in} \right) P[S_{i,m} = j \mid S_m = n]. \tag{7}$$

Naslednja zadeva, ki nas bo zanimala je $P[S_{i,m} = j, S_m = n]$. Zaradi neodvisnosti med S_{m-i} in $S_{i,m}$ in dejstva, da $S_{i,m} \sim S_i$, velja:

$$\begin{aligned} &P[S_{i,m} = j, S_m = n] = P[S_{i,m} = j, S_m = n] = \\ &= P[S_{i,m} = j, S_{m-i} = n - j] = P[S_i = j] P[S_{m-i} = n - j]. \end{aligned} \tag{8}$$

Zdaj lahko končno začnemo računati tisto, kar nas zanima:

$$P[S = n] = \sum_{m=1}^{\infty} P[S_m = n, N = m] = \sum_{m=1}^{k+\ell-1} P[S_m = n] P[N = m] + A_n \ , \tag{9}$$

kjer je

$$A_n = \sum_{m=k+\ell}^{\infty} P[S_m = n] P[N = m].$$

Vsoto torej razbijemo na dva dela in si natančneje ogledamo drugi del te vsote A_n. Velja

$$\begin{aligned} A_n &= \sum_{m=k+\ell}^{\infty} \sum_{i=1}^{\ell} \left(a_i + \frac{b_i}{m} \right) P[S_m = n] P[\widetilde{N}_i = m - i] = \\ &= \sum_{m=k+\ell}^{\infty} \sum_{i=1}^{\ell} \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in} \right) P[S_{i,m} = j \mid S_m = n] P[S_m = n] P[\widetilde{N}_i = m - i] = \\ &= \sum_{m=k+\ell}^{\infty} \sum_{i=1}^{\ell} \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in} \right) P[S_{i,m} = j, S_m = n] P[\widetilde{N}_i = m - i] = \\ &= \sum_{m=k+\ell}^{\infty} \sum_{i=1}^{\ell} \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in} \right) P[S_i = j] P[S_{m-i} = n - j] P[\widetilde{N}_i = m - i] = \\ &= \sum_{i=1}^{\ell} \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in} \right) P[S_i = j] \sum_{m=k+\ell}^{\infty} P[S_{m-i} = n - j] P[\widetilde{N}_i = m - i]. \end{aligned} \tag{10}$$

V drugo vrstico pridemo z uporabo predpostavke izreka (4), medtem ko za prehod v tretjo vrstico potrebujemo prej izpeljani rezultat (7). Iz četrte v peto vrstico uporabimo (8), vrstni red seštevanja pa lahko spremenimo zaradi konvergence vrste:

$$\sum_{m=k+\ell}^{\infty} P[S_{m-i} = n-j]P[\widetilde{N}_i = m-i] =$$
$$= \sum_{m=i}^{\infty} P[S_{m-i} = n-j]P[\widetilde{N}_i = m-i] =$$
$$= \sum_{z=0}^{\infty} P[S_z = n-j]P[\widetilde{N}_i = z] =$$
$$= \sum_{z=0}^{\infty} P[S_z = n-j, \widetilde{N}_i = z] = P[\widetilde{S}_i = n-j] < \infty.$$

Prva enakost sledi zaradi predpostavke (3) izreka 4.1, v nadaljevanju pa zamenjamo indeks h z $m+i$ in uporabimo neodvisnost.

Dobljeno vstavimo v (10) in dobimo

$$A_n = \sum_{i=1}^{\ell} \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in}\right) P[S_i = j]P[\widetilde{S}_i = n-j].$$

To zdaj vstavimo v (9) in sledi:

$$P[S=n] = \sum_{m=1}^{k+\ell-1} P[S_m = n]P[N=m] + \sum_{i=1}^{\ell} \sum_{j=0}^{n} \left(a_i + \frac{b_i j}{in}\right) P[S_i = j]P[\widetilde{S}_i = n-j],$$

kar zaključuje dokaz.

□

Opomba 4.2.

Trditev 3.3 je poseben primer izreka 4.1, kjer vzamemo $\ell = 1$. Tako se (6) prevede v

$$P[S=n] = P[S_k = n]P[N=k] + a_1 P[S_1 = 0]P[\widetilde{S}_1 = n]$$
$$+ \sum_{j=1}^{n} \left(a_1 + \frac{b_1 j}{n}\right) P[S_1 = j]P[\widetilde{S}_1 = n-j].$$

Zdaj izberemo enako porazdeljeni spremenljivki N in $\widetilde{N_1}$, kar implicira, da je $P[\widetilde{S}_1 = n] = P[S=n]$. Poleg tega po definiciji velja $S_1 = X_1 \sim X$, zato sledi, da je

$$P[S=n] = P[S_k = n]P[N=k] + a_1 P[X=0]P[S=n]$$
$$+ \sum_{j=1}^{n} \left(a_1 + \frac{b_1 j}{n}\right) P[X=j]P[S=n-j].$$

Izrazimo $P[S=n]$ in dobimo ravno (2), zato dokaz trditve 3.3 sledi iz dokaza izreka 4.1.

Sedaj je treba uporabiti zgornji izrek za numerično stabilna algoritma pri obeh posplošenih porazdelitvah N-ja.

5. Uporaba izreka pri ExtNegBin

Naj bo $k \in \mathbb{N}_0$, $\alpha \in (-k, -k+1)$ in $p \in (0,1)$. Dalje naj velja

$$N \sim ExtNegBin(\alpha - 1, k+1, p), \;\; \widetilde{N} \sim ExtNegBin(\alpha, k, p).$$

Potem lahko uporabimo izrek 4.1 za $\ell = 1$, kjer sta konstanti $a_1 = 0$ in

$$b_1 = (\alpha - 1)(1-p)\frac{p^{-\alpha} - \sum_{j=0}^{k-1}\binom{\alpha+j-1}{j}(1-p)^j}{p^{1-\alpha} - \sum_{j=0}^{k}\binom{\alpha+j-2}{j}(1-p)^j}.$$

Po izreku torej sledi, da je

$$P[S=0] = \begin{cases} 0 & ; P[X=0] = 0 \\ E[(P(X=0])^N] & ; \text{sicer} \end{cases},$$

in

$$P[S=n] = \frac{b_1}{n}\sum_{j=1}^{n} jP[X=j]P[\widetilde{S} = n-j], \quad n \in \mathbb{N}.$$

V zgornji uporabi izreka 4.1 nastopata dve slučajni spremeljivki N in $\widetilde{N}$, ki sta porazdeljeni posplošeno negativno binomsko, kjer se prva dva parametra razlikujeta za 1, p pa je enak pri obeh. S pomočjo tega pridemo do stabilnega algoritma za izračun porazdelitve slučajne vsote S.

5.1. Stabilni algoritem za ExtNegBin.

- Izvedi osnovni Panjerjev algoritem za $N \sim NegBin(\alpha + k, p)$, ki je za tako porazdeljen N numerično stabilen. Zdaj imamo torej porazdelitev za slučajno vsoto $\widetilde{S}_0$, kjer je število zahtevkov porazdeljeno $NegBin(\alpha+k, p) = ExtNegBin(\alpha + k, 0, p)$.
- Uporabi izrek 4.1 na zgoraj opisan način, kjer za $\widetilde{S}$ vzameš kar $\widetilde{S}_0$. Tako lahko stabilno izračunamo porazdelitev slučajne vsote $\widetilde{S}_1$, kjer je
$$N \sim ExtNegBin(\alpha + k - 1, 1, p).$$
- Uporabi izrek 4.1 na zgoraj opisan način, kjer za $\widetilde{S}$ vzameš kar $\widetilde{S}_1$. Tako lahko stabilno izračunamo porazdelitev slučajne vsote $\widetilde{S}_2$, kjer je
$$N \sim ExtNegBin(\alpha + k - 2, 2, p).$$
- Ponovi vajo še $(k-2)$ -krat, dokler ne prideš do porazdelitve slučajne vsote, kjer je
$$N \sim ExtNegBin(\alpha, k, p).$$

Opisali smo torej algoritem za stabilen izračun željene porazdelitve. Treba je povedati tudi to, da je v primerjavi z osnovnim algoritmom, ki je numerično nestabilen, ta algoritem časovno zahtevnejši za faktor k. Torej izgubimo na hitrosti izvedbe algoritma, a so rezultati bolj natančni, kar je pomembnejše.

Rezultati so bolj točni, saj ne pride do spremembe predznaka v členu $(a+\frac{bj}{n})$, ker je a enak 0 in lahko iz vsote $\frac{b}{n}$ tudi izpostavimo. Naslednji razlog se skriva v delu algoritma, kjer se dejansko izračuna $P[S=n]$. Za razliko od osnovnega Panjerjevega algoritma tu ni rekurzivne formule, saj pri računanju $P[S=n]$ ne nastopa nobena $P[S=i]$, kjer $i \leq n$, temveč le $P[\widetilde{S}=i]$, ki pa je stabilno izračunana. Ravno zaradi tega, ker ni rekurzije, pa se napake pri zaokroževanju, ki so neizogibne, ne kopičijo tako, kot je bilo to v osnovnem algoritmu.

Na naslednji strani je prikazan primer, ki zopet prikaže pomanjkljivosti osnovnega Panjerjevega algoritma pri posplošeni negativni binomski porazdelitvi in hkrati primerja rezultate, dobljene z uporabo zgornjega, numerično stabilnega algoritma.

Primer 5.1.
Naj bo $k \in \mathbb{N}$, $\epsilon, p \in (0,1)$, $\alpha = -k+\epsilon$ in $N \sim ExtNegBin(\alpha, k, p)$. Izberemo $\ell \in \mathbb{N} \setminus \{1,2\}$ in porazdelitev višine škodnih zahtevkov

$$X_i \sim X \sim \begin{pmatrix} 1 & \ell \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix},\ i=1,2,\dots\ .$$

Po definiciji N je $P[N<k]=0$. Torej mora biti vedno vsaj k zahtevkov. Zaradi tega in neodvisnosti N in X je

$$\begin{aligned} P[S=k] &= P[N=k, X_1=1,\dots,X_k=1] = \frac{P[N=k]}{2^k}, \\ P[S=k-1+\ell] &= \sum_{j=1}^{k} P[N=k, X_j=\ell, X_i=1 \text{ za } i \in \{1,\dots,k\}\setminus\{j\}] + \\ &+ P[N=k-1+\ell, X_1=1,...,X_{k-1+\ell}=1] = \frac{kP[N=k]}{2^k} + \frac{P[N=k-1+\ell]}{2^{k-1+\ell}}. \end{aligned}$$

Glede na to, da slučajna spremenljivka $S_k = X_1+...+X_k$ ne more zavzeti vrednosti $k+\ell$ in da je $P[X=0]=0$, se pri uporabi osnovnega Panjerjevega algoritma formula za izračun $P[S=k+\ell]$ prevede na

$$P[S=k+\ell] = \sum_{j=1}^{k+\ell} \left(a+\frac{bj}{n}\right) P[X_1=j]P[S=k+\ell-j].$$

Ker je člen $P[S=k+\ell-j]$ v zgornji vsoti neničeln le, ko je $j \in \{1,\ell\}$ in je posplošena negativna binomska porazdelitev v razredu $Panjer(q, q(\alpha-1), k)$, kjer je $q=1-p$, lahko nadaljujemo račun

$$P[S=k+\ell] = q\left(1+\frac{\alpha-1}{k+\ell}\right)\frac{P[S=k-1+\ell]}{2} + q\left(1+\frac{\alpha-1}{k+\ell}\ell\right)\frac{P[S=k]}{2}.$$

Ko vstavimo zgoraj izračunani verjetnosti $P[S=k]$ in $P[S=k-1+\ell]$ ter $\alpha=-k+\epsilon$, dobimo

$$\begin{aligned}
P[S=k+\ell] &= q\left(1+\frac{-k+\epsilon-1}{k+\ell}\right)\left(\frac{kP[N=k]}{2^{k+1}}+\frac{P[N=k+\ell]}{2^{k+\ell}}\right)+\\
&+q\left(1+\frac{-k\ell+\ell\epsilon-\ell}{k+\ell}\right)\frac{P[N=k]}{2^{k+1}}=\\
&=q\left(\frac{\ell+\epsilon-1}{k+\ell}\right)\left(\frac{kP[N=k]}{2^{k+1}}+\frac{P[N=k+\ell]}{2^{k+\ell}}\right)+\\
&+q\left(\frac{k-k\ell+\ell\epsilon}{k+\ell}\right)\frac{P[N=k]}{2^{k+1}}.
\end{aligned}$$

Želimo pokazati, da se dva poljubno majhna člena odštejeta, zato velja iz zadnjega oklepaja v prvem členu vsote izpostaviti k. Dobimo

$$\begin{aligned}
P[S=k+\ell] &= q\left(\frac{-k+k\ell+k\epsilon}{k+\ell}\right)\left(\frac{P[N=k]}{2^{k+1}}+\frac{P[N=k+\ell]}{k2^{k+\ell}}\right)+\\
&+q\left(\frac{k-k\ell+\ell\epsilon}{k+\ell}\right)\frac{P[N=k]}{2^{k+1}}=\\
&=q\left(\frac{-k+k\ell+k\epsilon}{k+\ell}\right)\frac{P[N=k+\ell]}{k2^{k+\ell}}+\\
&+q\frac{P[N=k]}{2^{k+1}}\left(\frac{-k+k\ell+k\epsilon}{k+\ell}-\frac{-k+k\ell-\ell\epsilon}{k+\ell}\right).
\end{aligned}$$

V zadnjem delu izračuna se vidi, da pride do odštevanja dveh števil, ki sta poljubno blizu. Manjši kot je ϵ, bližje sta si. Poleg tega opazimo, da je prvi člen v primerjavi z drugim, ki je problematičen, praktično zanemarljiv, saj je pomnožen z $\frac{1}{2^{k+\ell}}$. Drugi člen pa je pomnožen le z $\frac{1}{2^{k+1}}$, torej večji kot je ℓ, manj pomemben je prvi člen. Za boljšo predstavo si poglejmo, kaj se zgodi pri naslednjih parametrih. Naj bo

$$\epsilon = 10^{-4}, k=1, \ell=5 \text{ in } p=\frac{1}{10}.$$

S spodnjo kodo v programu R izračunamo verjetnost $P[S=6]$.

```
k <- 1
alfa <- -k + 10**(-4)
p <- 0.1
l <- 5
q <- rep(0,k-1)
for(n in k:100){
  suma <- 0
  for(i in 0:(k-1)){
    suma <- suma+ choose(alfa+i-1,i)*(1-p)**i}
  q <- c(q,choose(alfa+n-1,n)*(1-p)**n/(p**(-alfa) - suma))}
ver <- (1-p)*q[1]/(2**(k+1))*((-k+k*l+k*10**(-4))/(k+l)-
                      (-k+k*l-l*10**(-4))/(k+l))+
                      (1-p)*(-k+k*l+k*10**(-4))/(k+l)*
                      q[k+l]/(k*2**(k+l))
```

Dobimo:

$$P[S=6]=0.00002251678.$$

Na naslednji strani je priložena tabela verjetnosti, ki jih dobimo, če uporabimo numerično stabilen algoritem 5.1.

Višina škod	Verjetnost
1	0.49996279266
2	0.00001124916
3	0.00000168754
4	0.00000037971
5	0.49996289519
6	0.00002252908
7	0.00000507525
8	0.00000152220
9	0.00000051380
10	0.00001143414

Primerjamo izračunano vrednost $P[S=6]$ s to v tabeli in vidimo, da pride do odstopanj. Razlog je seveda že večkrat omenjeno odštevanje majhnih števil.

6. Uporaba izreka pri ExtLog

Naj bo $k \in \mathbb{N} \setminus \{1\}$ in $q \in (0,1)$. Dalje naj bosta

$$N \sim ExtLog(k+1,q),\ \widetilde{N} \sim ExtLog(k,q)$$

Potem vzamemo $\ell = 1$ in uporabimo izrek 4.1. Konstanti sta $a_1 = 0$ in

$$b_1 = (k+1)q\frac{\sum_{\ell=k}^{\infty}\binom{\ell}{k}^{-1}q^\ell}{\sum_{\ell=k+1}^{\infty}\binom{\ell}{k+1}^{-1}q^\ell}.$$

Po izreku sledi, da je

$$P[S=0]=\begin{cases} 0 & ; P[X=0]=0 \\ E[(P(X=0])^N] & ; \text{sicer} \end{cases} \tag{11}$$

in

$$P[S=n]=\frac{b_1}{n}\sum_{j=1}^{n} jP[X=j]P[\widetilde{S}=n-j],\ n\in\mathbb{N}. \tag{12}$$

Podobno kot prej imamo tudi tu dve slučajni spremenljivki N in $\widetilde{N}$, ki sta porazdeljeni posplošeno logaritemsko, kjer se prvi parameter razlikuje za 1. Spodaj je algoritem, ki stabilno izračuna porazdelitev slučajne vsote, kjer je
$N \sim ExtLog(k,q)$.

6.1. **Stabilni algoritem za ExtLog.**

- Izvedi osnovni Panjerjev algoritem za $N \sim Log(q)$, ki je za tako porazdeljen N numerično stabilen. Zdaj imamo torej porazdelitev za slučajno vsoto $\widetilde{S}_0$, kjer je število zahtevkov porazdeljeno $N \sim Log(q) = ExtLog(1, q)$.
- Uporabi zgoraj opisano uporabo izreka 4.1, kjer za $\widetilde{S}$ vzameš kar $\widetilde{S}_0$. Tako lahko stabilno izračunamo porazdelitev slučajne vsote $\widetilde{S}_1$, kjer je
$$N \sim ExtLog(2, q).$$
- Uporabi zgoraj opisano uporabo izreka 4.1, kjer za $\widetilde{S}$ vzameš kar $\widetilde{S}_1$. Tako lahko stabilno izračunamo porazdelitev slučajne vsote $\widetilde{S}_2$, kjer je
$$N \sim ExtLog(3, q).$$
- Ponovi vajo še $(k-2)$ -krat, dokler ne prideš do porazdelitve slučajne vsote, kjer je $N \sim ExtLog(k, q)$.

Ta algoritem je zelo podoben tistemu za posplošeno negativno binomsko porazdelitev, posledično je tudi spodnji komentar praktično enak. Algoritem seveda izračuna porazdelitev S numerično stabilno, toda je časovno zahtevnejši.
Uporabimo isti glavni izrek kot prej, zato se tudi tu izognemo spremembi predznaka v problematičnem členu. Prav tako ne gre za rekurzivno formulo, kar prepreči kopičenje zaokrožitvenih napak.

7. Uporaba v praksi

Že v uvodu sem nekaj stavkov namenil temu, kje bi Panjerjev algoritem lahko bil uporaben. Pomembno je, da ima finančna institucija ali podjetje potencialne stroške oziroma obveznosti. Ni seveda nujno, da do njih pride, v primeru zavarovalnic do odškodninske poravnave običajno sploh ne pride. Druga stvar, ki mora veljati je, da so višine teh stroškov vsaj do neke mere podobne. Ne moremo v isti koš metati terjatev, ki nastanejo pri avtomobilskih zavarovanjih in tistih pri življenjskih zavarovanjih. Tudi število škodnih zahtevkov, ki pridejo na naslov zavarovalnic, je odvisno od tega, o kakšnem tipu zavarovanja je govora.

7.1. **Izbira porazdelitve N v praksi.**

Izmed šestih verjetnostnih porazdelitev, ki spadajo v Panjerjev razred, so tri primerne za uporabo že pri osnovnem Panjerjevem algoritmu. S tem mislim na to, da algoritem izračuna porazdelitev slučajne vsote S numerično stabilno. To so Poissonova, negativna binomska in logaritemska. Logaritemska se v praksi praktično ne uporablja, medtem ko sta preostali dve v aktuarstvu bolj popularni.
Ostale tri porazdelitve, kjer osnovni Panjerjev algoritem ni najboljša izbira zaradi numeričnih težav, ki nastopijo, pa so posplošena negativna binomska, posplošena logaritemska in binomska. Za obe posplošeni porazdelitvi smo zgoraj videli, kako se malce prilagodi algoritem, da je izračun porazdelitve S numerično stabilen. Treba je

komentirati tudi dejstvo, da sta ti dve porazdelitvi zelo specifični, saj je skoraj vsa verjetnost skoncentrirana v k. Torej je zelo malo verjetno, da dobimo več zahtevkov kot k. To najbrž pri običajnih zavarovalnicah ni smiselna izbira porazdelitve N, je pa bolj uporabno v pozavarovalništvu.

Preostala je še ena porazdelitev, za katero je algoritem numerično problematičen, in sicer binomska. Binomska porazdelitev pa v aktuarstvu tako ali tako ni najbolj popularna za modeliranje v kolektivnem modelu. Prva pomanjkljivost te porazdelitve je, da je omejena z n, česar si v praksi včasih ne želimo. Poleg tega je standardni odklon majhen in zato ta izbira porazdelitve za N ni najbolj primerna.

7.2. **Izbira porazdelitve X v praksi.**

Najprej je treba razčistiti predpostavko, da je $X \in \mathbb{N}_0$. V realnosti so seveda višine škod porazdeljene zvezno, vendar pa za uporabo Panjerjevega algoritma to predpostavko potrebujemo. Zato se v praksi znajdemo tako, da zvezno slučajno spremenljivko diskretiziramo z metodo zaokroževanja.

7.2.1. *Diskretizacija z metodo zaokroževanja.*

Naj bo Y zvezna slučajna spremenljivka, ki jo želimo diskretizirati in X diskretna slučajna spremenljivka, ki je rezultat diskretizacije. Izberimo si $h > 0$ in določimo

$$\begin{aligned} P[X = jh] &= P\left[\left(j - \frac{1}{2}\right)h \leq Y < \left(j + \frac{1}{2}\right)h\right] \\ &= F_Y\left(\left(j + \frac{1}{2}\right)h\right) - F_Y\left(\left(j - \frac{1}{2}\right)h\right) \end{aligned}$$

za $j \in \mathbb{N}_0$, kjer je F_Y kumulativna porazdelitvena funkcija spremenljivke Y.

V našem primeru si seveda izberemo $h = 1$. Zdaj je treba le določiti, katera zvezna slučajna spremenljivka bo predstavljala višine škod. Nekako najbolj se uporabljajo naslednje:

- $Paretova(\alpha, x_m)$, kjer za parametra velja $\alpha > 0$ in $x_m > 0$, z gostoto
$$f(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}},\ x > x_m$$
in upanjem $\frac{\alpha x_m}{\alpha+1}$.

- $Eksponentna(\lambda)$, kjer za parameter velja $\lambda > 0$, z gostoto
$$f(x) = \lambda e^{-\lambda x},\ x > 0$$
in upanjem $\frac{1}{\lambda}$.

- $Weibullova(k, \lambda)$, kjer za parametra velja $\lambda > 0$ in $k > 0$, z gostoto
$$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-\frac{x}{\lambda}^k},\ x > 0$$
in upanjem $\lambda\Gamma(1 + \frac{1}{k})$.

Naslednje vprašanje, ki se postavlja je, kako določiti parametre porazdelitev. Marsikaj je seveda odvisno od tega, kakšna zavarovanja so v igri oziroma bolj pomembno, kako visoke so odškodnine. Najpogosteje zavarovalnice uporabijo podatke iz preteklosti za opazovan tip zavarovanja in ocenijo parametre z znano metodo največjega verjetja.

Primer 7.1. Uporaba na konkretnih podatkih

Zdi se mi pomembno, da zna finančni matematik pridobljeno znanje uporabiti tudi v praksi in to je glavni namen tega zgleda.

Na spletnih straneh agencije republike Slovenije za okolje sem našel poročilo o analizi odškodninskih zahtevkov za škodo, ki so jo povzročile živali zavarovanih prosto živečih živalskih vrst v letih 2007-2012. Zbral sem podatke o škodi, ki jo je v teh letih povzročil dihur. Bolj natančno, pogledal sem, koliko in kako visoki so bili posamezni odškodninski zahtevki zaradi škode dihurja.

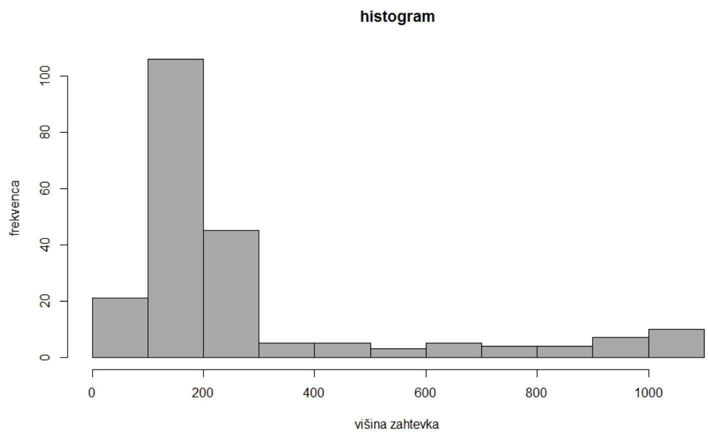
Podatkov je bilo na voljo 215, zato sem za predstavitev le teh raje podal števila in vsote škod v posameznih letih, kar je prikazano v spodnji tabeli.

Leto	Število zahtevkov	Letna škoda
2007	17	4750.02 €
2008	38	14680.57 €
2009	44	20888.46 €
2010	41	9956.05 €
2011	24	3254.83 €
2012	51	6047.32 €

Na podlagi teh podatkov je treba napovedati, kaj se bo zgodilo naslednje leto. Najprej se je treba odločiti, kako bo porazdeljeno število zahtevkov. Odločil sem se za Poissonovo porazdelitev in za oceno λ po metodi največjega verjetja za Poissonovo porazdelitev vzamem povprečje. Prva predpostavka je torej, da je

$$N \sim Pois\left(\frac{17+38+44+41+24+51}{6}\right) = Pois(35.8333).$$

Sedaj je treba določiti porazdelitev X. Po pregledu histograma



in grafa gostote Paretove porazdelitve sem se odločil za slednjo. Na razpolago je 215 podatkov, s pomočjo katerih sem po metodi največjega verjetja za Paretovo porazdelitev ocenil parametra α in x_m. Najprej je treba izpeljati funkcijo verjetja

$$L(\alpha, x_m) = \prod_{i=1}^{n} \frac{\alpha x_m^{\alpha}}{x_i^{\alpha+1}},\ 0 < x_m \leq \min_i\{x_i\},\ \alpha > 0,$$

kjer sta x_m in α parametra porazdelitve, $\{x_i\}_{i=1}^n$ pa so konkretni podatki. Opazim, da je vrednost funkcije verjetja z naraščajočim x_m vedno večja, zato za x_m vzamem največ, kar lahko. Torej je ocena za x_m enaka

$$\hat{x}_m = \min_i\{x_i\}.$$

Sedaj fiksiram ta parameter in ga vstavim v funkcijo verjetja. Za oceno α se bo treba bolj potruditi in odvajati. Ker je L nenegativna funkcija, jo lahko logaritmiram in zaradi monotonosti logaritemske funkcije ostanejo točke, kjer so doseženi ekstremi, enake. Funkcija, katere ekstrem je treba najti, je torej

$$\log(L(\alpha, \hat{x}_m)) = \sum_{i=1}^{n} \log(\frac{\alpha \hat{x}_m^{\alpha}}{x_i^{\alpha+1}}) =$$

$$= n\log(\alpha) + \alpha n \log(\hat{x}_m) - (\alpha+1)\sum_{i=1}^{n}\log(x_i).$$

Poiskati je treba stacionarne točke, zato odvajam in dobim

$$\frac{n}{\alpha} + nlog(\hat{x}_m) - \sum_{i=1}^{n}\log(x_i).$$

Zdaj poiščem α, pri katerem je odvod enak nič in dobim

$$\alpha = \frac{n}{\sum_{i=1}^{n}\log(x_i) - n\log(\hat{x}_m)} = \frac{n}{\sum_{i=1}^{n}\log(\frac{x_i}{\hat{x}_m})}.$$

Preverim še, da je v edini stacionarni točki res maksimum, zato še enkrat odvajam. Drugi odvod je enak $-\frac{n}{\alpha^2}$ in je negativen v vsaki, tudi stacionarni točki, kar pomeni, da je to res maksimum. Ocena za α je torej

$$\hat{\alpha} = \frac{n}{\sum_{i=1}^{n}\log(\frac{x_i}{\hat{x}_m})}.$$

Vstavim podatke in ocenim parametra v programu R:

```
podatki <- t(read.table("podatki.txt"))
x_m <- min(podatki)
alfa <- length(podatki)/sum(log(podatki/x_m))
```

Dobim

$$\hat{x}_m = 79.1 \text{ in } \hat{\alpha} = 1.048.$$

Zdaj sta znani obe porazdelitvi $N \sim Pois(35.8333)$ in $X \sim Pareto(1.048, 79.1)$. Ker želim uporabiti osnovni Panjerjev algoritem, je treba najprej diskretizirati X po metodi zaokroževanja. V R-u naložim paket *actuar*, ki ima to že vgrajeno.

```
h <- 1
n <- 100000/h
diskretno = discretize(ppareto1(x,alfa,x_m),
                       from = 0,  to = n*h, step = h,
                       method="rounding")
```

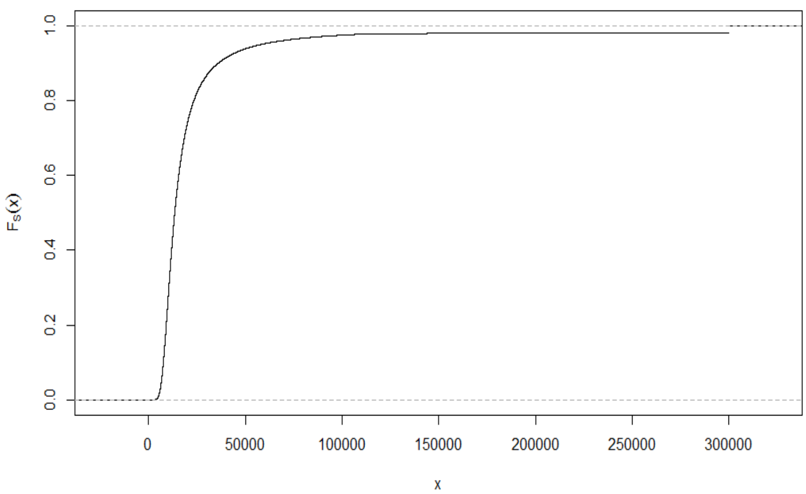
Tako dobim verjetnostno funkcijo diskretizirane slučajne spremenljivke, in sicer za $n \in \{0, 1, \ldots, 100000\}$. Nekje je bilo treba določiti zgornjo mejo in izbral sem 100000. S tem ni storjene večje napake, saj je največji zahtevek enak 1200. Poleg tega velja

$$P[X \leq 100000] = 0.9994402$$

in ker je ta verjetnost tako velika, se mi ni zdelo smiselno še povečevati meje. V tem paketu je prav tako vgrajena funkcija, ki na več načinov izračuna porazdelitev S. Izbrati je treba metodo *recursive*, ki uporablja Panjerjev algoritem. Maksimalno število iteracij sem postavil na 300000.

```
porazdelitevS <- aggregateDist("recursive",
                               model.freq = "poisson",
                               model.sev = diskretno,
                               lambda = 35.8333,
                               maxit = 300000,x.scale=h)
```

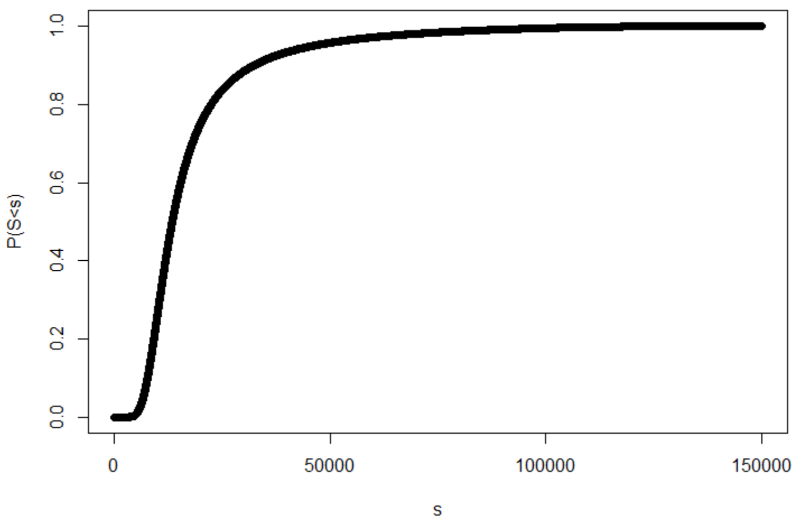
Ko se program izvrši, izvem, da mu ni uspelo izračunati porazdelitve do konca, saj bi potreboval več iteracij. Kljub temu je vsota vektorja, ki predstavlja verjetnostno funkcijo enaka 0.98014, tako da nisem povečeval števila iteracij, saj bi trajalo predolgo. Spodaj je graf porazdelitvene funkcije slučajne vsote S.



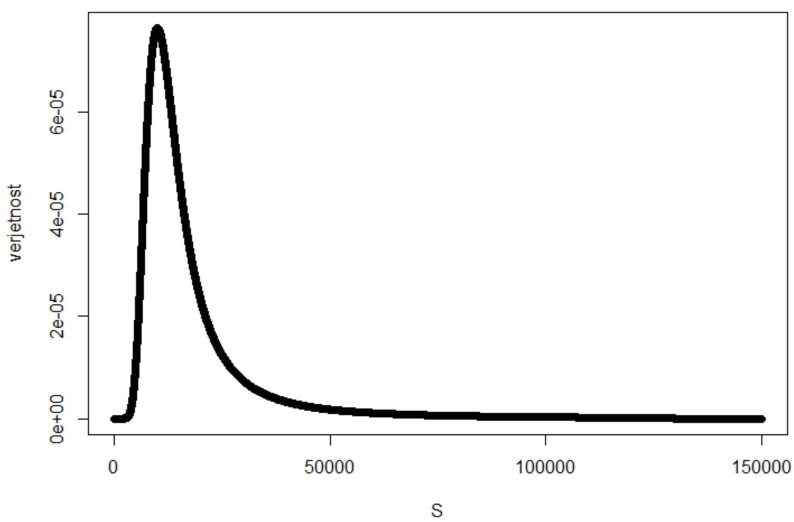
Opazil sem, da kumulativna porazdelitvena funkcija med vrednostima 150000 in 300000 praktično ne narašča. To pomeni, da je velika večina tiste verjetnosti 98%, ki jo algoritem naračuna, zbrane do vrednosti 150000. Velja namreč, da je $P[S \in (150000, 300000)] = 0.000178$. Zato tistih dveh odstotkov, ki manjkata, ne bi dobili s povečanjem števila iteracij, temveč ta napaka izhaja od drugje. Zaradi teh razlogov se mi zdi smiselno verjetnostno funkcijo normirati in predpostaviti, da S zavzame le vrednosti do 150000. Glede na to, da je v podatkih največji škodni zahtevek enak 1200 in da jih je večina celo manjših od 300, ta predpostavka ni tako napačna.

```
meja <- porazdelitevS(150000)
FS1 <- diff(porazdelitevS)/meja
x <- 1:150000
FS <- FS1[x]#verjetnostna funkcija v vektorju
```

Spodaji je graf porazdelitvene funkcije po normiranju.



In še graf verjetnostne funkcije po normiranju.



Iz teh grafov in verjetnostne funkcije, ki jo imamo na voljo, lahko zavarovalnici pomagamo, da ustrezno količino denarja rezervira za obveznosti naslednje leto in višek naloži ter še nekaj zasluži. Najprej iz grafov opazimo, da je večina verjetnosti zbrane tam do 30000. Potem gremo računati karakteristike S. Prva stvar, ki nas zanima je matematično upanje in to je enako 17917.81.

```
upanje <- x  %*% FS
```

Če pogledamo tabelo podatkov, opazimo, da je bila škoda večja od upanja le leta 2009. To nam da idejo, da si pogledamo verjetnost, da bo škode manj od pričakovane škode:

$$P[S < 17917.81] = 0.697002.$$

Podatki se s tem skladajo. Nato opazimo, da se škode v štirih izmed šestih let precej razlikujejo od upanja, zato izračunamo standardni odklon, ki je enak 14719.65.

```
odklon <- sqrt((x**2) %*% FS - upanje**2)
```

Standardni odklon je torej velik, zato nas prejšnja ugotovitev ne čudi. Hkrati opazimo, da je bila škoda v vseh izmed teh štirih let manjša od upanja. Sumimo na pozitivno asimetrijo, kar drži. Asimetrija je namreč enaka 3.185885.

```
asimetrija <- ((x**3)%*% FS -
                    3*upanje*x**2 %*% FS +
                    3*upanje**2*x%*% FS -
                    upanje**3)/(odklon**3)
```

Lahko si pogledamo tudi znano mero tveganja *value at risk* pri stopnji tveganja β, ki nam pove, da bo škode z verjetnostnjo $1-\beta$ manj od $\mathrm{VaR}_{1-\beta}$.

$$\mathrm{VaR}_{0.95} = 46021, \quad \mathrm{VaR}_{0.90} = 32286, \quad \mathrm{VaR}_{0.80} = 22399$$

Vsega, kar je zgoraj izračunano, pa se seveda ne sme jemati povsem resno. Problematično je že to, da imamo zelo majhen vzorec le šestih let. Tudi predpostavke o porazdelitvah niso točne, o neodvisnosti bi prav tako lahko razpredali, saj je škoda, ki jo povzročijo dihurji odvisna od recimo vremenskih razmer, ki pa se skozi leta razlikujejo. Poleg tega ne gre zanemariti niti tiste napake dveh odstotkov in normiranja. Kljub vsemu temu pa menim, da je za nek občutek o tem, kaj se bo dogajalo v prihodnosti, ta zgled bil vendarle koristen.

Literatura

[1] S. Gerhold, U. Schmock, R. Warnung. *A generalization of Panjer's recursion and numerically stable risk aggregation*, Finance and Stochastics 14(2010), 81-128.

[2] U. Schmock, *Approximation and Aggregation of Risks by Variants of Panjer's Recursion*, verzija 8. 2. 2013, [ogled 12. 8. 2014], dostopno na `http://www.fam.tuwien.ac.at/~schmock/slides/Variants-Panjer-Slides.pdf`.

[3] H. Panjer, *Recursive evaluation of a family of compound distributions*, Astin Bulletin Volume 12 Number 1 (1981), 22-26.

[4] P. Embrechts, M. Frei. *Panjer recursion versus FFT for compound distributions*, verzija 1. 7. 2010, [ogled 8. 8. 2014], dostopno na `http://www.math.ethz.ch/~embrecht/ftp/PanjerVsFFTcorrected.pdf`.

[5] M. Fackler. *Panjer class united - one formula for the Poisson, Binomial and Negative Binomial distribution*, [ogled 10. 8. 2014], dostopno na `http://www.actuaries.org/ASTIN/Colloquia/Helsinki/Papers/S7_13_Fackler.pdf`.

[6] G. Willmot, *Sundt and Jewell's family of discrete distributions*, Astin Bulletin Volume 18 Number 1 (1988), 17-29.

[7] H. Panjer, G. Willmot. *Computational aspects of recursive evaluation of compound distributions*, Insurance: Mathematics and Economics, 5:113-116.

[8] P. Antal. *Quantitative Methods in Reinsurance*, verzija 8. 4. 2003, [ogled 8. 8. 2014], dostopno na `http://www.math.ethz.ch/finance/misc/MathMethodsReinsurance.pdf`.