Computation of the Aggregate Claim Amount Distribution Using R and actuar

Vincent Goulet, Ph.D.

UNIVERSITÉ LAVAL

Faculté des sciences et de génie
École d'actuariat

Actuarial Risk Modeling Process

1. Model costs at the individual level
   $\rightarrow$ Modeling of loss distributions
2. Aggregate risks at the collective level
   $\rightarrow$ Risk theory
3. Determine revenue streams
   $\rightarrow$ Ratemaking
4. Evaluate solvability of insurance portfolio
   $\rightarrow$ Ruin theory

What actuar Is

- A package providing additional Actuarial Science functionality to the R statistical system
- Distributed through the *Comprehensive R Archive Network* (CRAN)
- Currently provides:
  - 17 additional probability distributions
  - loss modeling facilities
  - aggregate claim amount calculation
  - fitting of credibility models
  - ruin probabilities and related quantities
  - simulation of compound hierarchical models

Yes But… Why R?

- Compare

  `x <- matrix(2, 3, 10:15)`   vs   `x ← 2 3ρ9 + ι6`

- Multi-platform
- Interactive
- State-of-the-art statistical procedures, random number generators and graphics

Collective Risk Model

- Let

$$\begin{aligned} S &: \text{aggregate claim amount} \\ N &: \text{number of claims (frequency)} \\ C_j &: \text{amount of claim } j \text{ (severity)} \end{aligned}$$

- We have the random sum

$$S = C_1 + \cdots + C_N$$

- We want to find

$$\begin{aligned} F_S(x) &= \Pr[S \le x] \\ &= \sum_{n=0}^{\infty} \Pr[S \le x | N = n] \Pr[N = n] \\ &= \sum_{n=0}^{\infty} F_C^{*n}(x) \Pr[N = n] \end{aligned}$$

Collective Risk Model

- Let

$$\begin{aligned} S &: \text{aggregate claim amount} \\ N &: \text{number of claims (frequency)} \\ C_j &: \text{amount of claim } j \text{ (severity)} \end{aligned}$$

- We have the random sum

$$S = C_1 + \cdots + C_N$$

- We want to find

$$\begin{aligned} F_S(x) &= \Pr[S \leq x] \\ &= \sum_{n=0}^{\infty} \Pr[S \leq x | N = n] \Pr[N = n] \\ &= \sum_{n=0}^{\infty} F_C^{*n}(x) \Pr[N = n] \end{aligned}$$

Collective Risk Model

- Let

$$\begin{aligned} S &: \text{aggregate claim amount} \\ N &: \text{number of claims (frequency)} \\ C_j &: \text{amount of claim } j \text{ (severity)} \end{aligned}$$

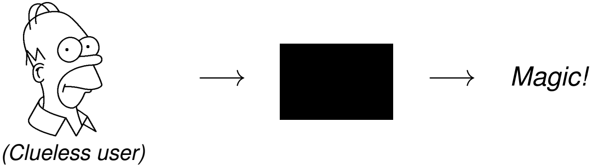
- We have the random sum

$$S = C_1 + \cdots + C_N$$

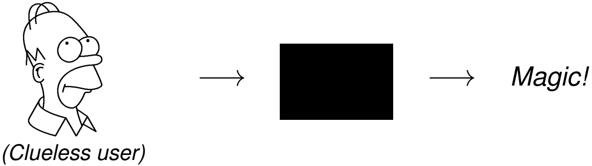
- We want to find

$$\begin{aligned} F_S(x) &= \Pr[S \leq x] \\ &= \sum_{n=0}^{\infty} \Pr[S \leq x | N = n] \Pr[N = n] \\ &= \sum_{n=0}^{\infty} F_C^{*n}(x) \Pr[N = n] \end{aligned}$$

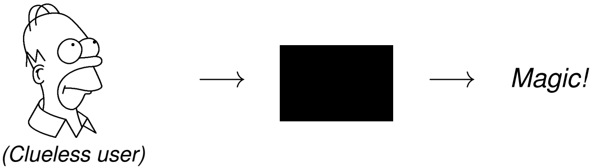
What actuar Tries Not To Be



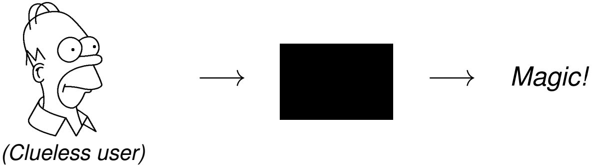
What actuar Tries Not To Be



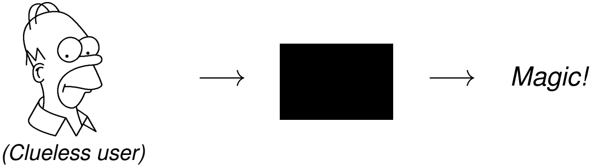
What actuar Tries Not To Be



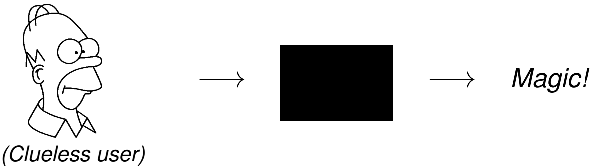
What actuar Tries Not To Be



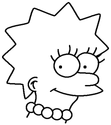
What actuar Tries Not To Be



What actuar Tries Not To Be



What We're Presenting Here Today



(Insightful user)

$\longrightarrow$ `aggregateDist()` $\longrightarrow$ $F_S(x)$

How We Can Tackle the Problem

1. Carry out the convolutions $F_C^{*k}(x)$ for $k = 0, 1, 2, \ldots$
2. Use a Normal approximation

$$F_S(x) \simeq \Phi\left(\frac{x - \mu_S}{\sigma_S}\right)$$

3. Use the Normal Power II approximation

$$F_S(x) \simeq \Phi\left(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x - \mu_S}{\sigma_S}}\right)$$

4. Use simulation:

$$F_S(x) \simeq F_n(x) = \frac{1}{n}\sum_{j=1}^{n} I\{x_j \leq x\}$$

How We Can Tackle the Problem

1. Carry out the convolutions $F_C^{*k}(x)$ for $k = 0, 1, 2, \ldots$ ✓
2. Use a Normal approximation

$$F_S(x) \simeq \Phi\left(\frac{x - \mu_S}{\sigma_S}\right)$$

3. Use the Normal Power II approximation

$$F_S(x) \simeq \Phi\left(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x - \mu_S}{\sigma_S}}\right)$$

4. Use simulation:

$$F_S(x) \simeq F_n(x) = \frac{1}{n}\sum_{j=1}^{n} I\{x_j \leq x\}$$

How We Can Tackle the Problem

1. Carry out the convolutions $F_C^{*k}(x)$ for $k = 0, 1, 2, \ldots$ ✓
2. Use a Normal approximation ✓

$$F_S(x) \simeq \Phi\left(\frac{x - \mu_S}{\sigma_S}\right)$$

3. Use the Normal Power II approximation

$$F_S(x) \simeq \Phi\left(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x - \mu_S}{\sigma_S}}\right)$$

4. Use simulation:

$$F_S(x) \simeq F_n(x) = \frac{1}{n}\sum_{j=1}^{n} I\{x_j \leq x\}$$

How We Can Tackle the Problem

1. Carry out the convolutions $F_C^{*k}(x)$ for $k = 0, 1, 2, \ldots$ ✓
2. Use a Normal approximation ✓

$$F_S(x) \simeq \Phi\left(\frac{x - \mu_S}{\sigma_S}\right)$$

3. Use the Normal Power II approximation ✓

$$F_S(x) \simeq \Phi\left(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x - \mu_S}{\sigma_S}}\right)$$

4. Use simulation:

$$F_S(x) \simeq F_n(x) = \frac{1}{n}\sum_{j=1}^{n} I\{x_j \leq x\}$$

How We Can Tackle the Problem

1. Carry out the convolutions $F_C^{*k}(x)$ for $k = 0, 1, 2, \ldots$ ✓
2. Use a Normal approximation ✓

$$F_S(x) \simeq \Phi\left(\frac{x - \mu_S}{\sigma_S}\right)$$

3. Use the Normal Power II approximation ✓

$$F_S(x) \simeq \Phi\left(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x - \mu_S}{\sigma_S}}\right)$$

4. Use simulation: ✓

$$F_S(x) \simeq F_n(x) = \frac{1}{n}\sum_{j=1}^{n} I\{x_j \leq x\}$$

Most Commonly Used Method

5 Recursive method (Panjer algorithm):

$$f_S(x) = \frac{1}{1 - af_C(0)} \Bigg[(p_1 - (a+b)p_0) f_C(x) + \sum_{y=1}^{\min(x,m)} (a + by/x) f_C(y) f_S(x-y) \Bigg]$$

with

$$f_S(0) = P_N(f_C(0))$$

Most Commonly Used Method

5 Recursive method (Panjer algorithm): ✓

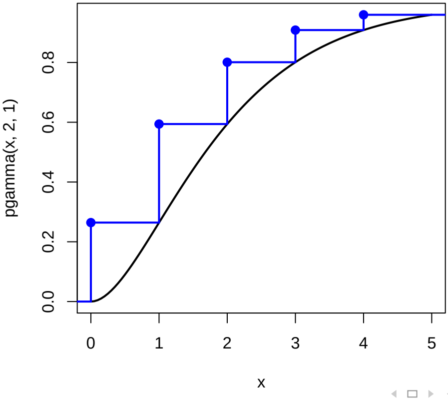
$$f_S(x) = \frac{1}{1 - af_C(0)} \Big[(p_1 - (a+b)p_0) f_C(x) + \sum_{y=1}^{\min(x,m)} (a + by/x) f_C(y) f_S(x-y) \Big]$$

with

$$f_S(0) = P_N(f_C(0))$$

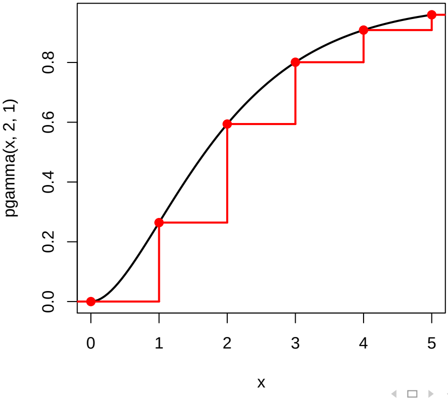
Discretization of Continuous Distributions

```
> discretize(pgamma(x, 2, 1), from = 0, to = 5,
             method = "upper")
```



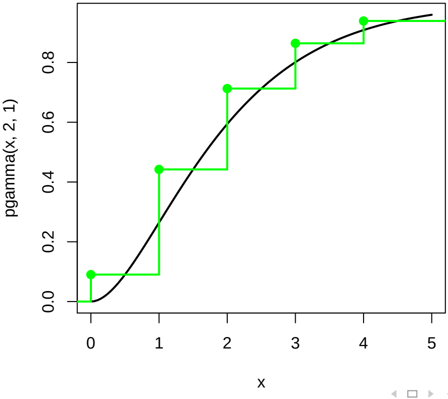
Discretization of Continuous Distributions

```
> discretize(pgamma(x, 2, 1), from = 0, to = 5,
             method = "lower")
```



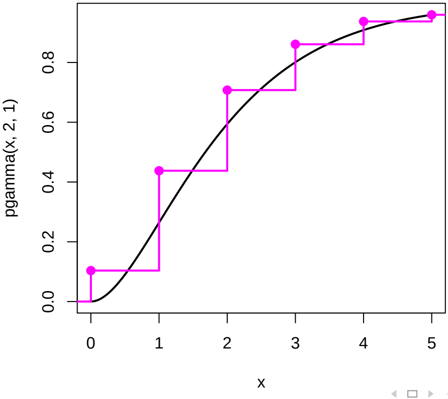
Discretization of Continuous Distributions

```
> discretize(pgamma(x, 2, 1), from = 0, to = 5,
             method = "rounding")
```



Discretization of Continuous Distributions

```
> discretize(pgamma(x, 2, 1), from = 0, to = 5,
             method = "unbiased",
             lev = levgamma(x, 2, 1))
```



Computing the Aggregate Claim Amount Distribution

- `aggregateDist()` is the unified interface to all 5 supported methods
- Computer intensive calculations are done in C
- Output is a **function** to compute $F_S(x)$ in any x
- R methods to plot and compute related quantities

Example

Assume

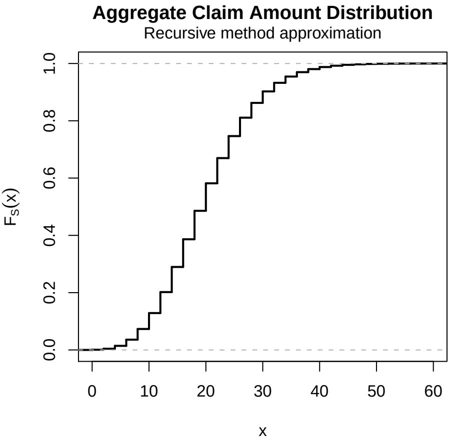
$$N \sim \text{Poisson}(10)$$
$$C \sim \text{Gamma}(2, 1)$$

```
> fx <- discretize(pgamma(x, 2, 1), from = 0,
                   to = 22, step = 2,
                   method = "unbiased",
                   lev = levgamma(x, 2, 1))
```

```
> Fs <- aggregateDist("recursive",
                      model.freq = "poisson",
                      model.sev = fx,
                      lambda = 10, x.scale = 2)
```

Example (continued)

```
> plot(Fs)
```



Example (continued)

```
> summary(Fs)

Aggregate Claim Amount Empirical CDF:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.00000 14.00000 20.00000 19.99996 26.00000 74.00000

> knots(Fs)

 [1]  0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32
[18] 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66
[35] 68 70 72 74

> Fs(c(10, 15, 20, 70))

[1] 0.1287553 0.2896586 0.5817149 0.9999979
```

Example (continued)

```
> mean(Fs)

[1] 19.99996

> VaR(Fs)

90% 95% 99%
 30  34  42

> TVaR(Fs)

     90%      95%      99%
35.99043 39.56933 46.97385
```

One more thing...

What If Recursions Do Not Start?

- For example, in the Compound Poisson case

$$f_S(0) = e^{-\lambda(1-f_C(0))}$$

- If λ is large, $f_S(0) = 0$ numerically
- One solution:
  1. divide λ by 2^n
  2. convolve resulting distribution n times with itself

Example

```
> Fsc <- aggregateDist("recursive",
                       model.freq = "poisson",
                       model.sev = fx,
                       lambda = 5, convolve = 1,
                       x.scale = 2)

> summary(Fsc)

Aggregate Claim Amount Empirical CDF:
     Min.   1st Qu.    Median      Mean   3rd Qu.
  0.00000  14.00000  20.00000  19.99997  26.00000
     Max.
108.00000
```

Concluding Remarks

- See `cran.r-project.org/package=actuar` for the package
- Package *vignettes* provide complete documentation
- Please cite the software in publications:

```
> citation(package = "actuar")

To cite actuar in publications use:

  C. Dutang, V. Goulet and M. Pigeon (2008).
  actuar: An R Package for Actuarial Science.
  Journal of Statistical Software, vol. 25, no.
  7, 1-37. URL http://www.jstatsoft.org/v25/i07
  [...]
```

- Contribute!