用 R 也能做精算—actuar 包学习笔记（三）

李皞
（中国人民大学 统计学院 风险管理与精算）

3 风险理论

本部分主要介绍风险理论中的聚合风险模型。在机动车保险中，对于一辆或一批机动车，其每年发生的事故次数N服从一个离散分布，每次事故的损失金额X服从一个离连续分布。那么，这一年总的损失额S可以表示为：

$$S = X_1 + X_2 + \ldots + X_N \tag{1}$$

可以看出S是一个随机和，我们把事故次数N的分布称作索赔频率分布（frequency distribution），每次损失额X的分布称作索赔强度分布（severity distribution），S的分布称为复合分布（compound distribution）。

上一小节讲如何估计分布的参数，假设我们已经将频率分布和强度分布的参数估计出来了，那么现在的问题就是如何得到总损失额S的分布，事实上，就保险公司的整体运营来讲，精算师可能更关心这个分布。对于S的分布，我们有：

$$\begin{aligned} F_S(x) &= P(S \leq x) \\ &= \sum_{n=0}^{\infty} P(S \leq x | N = n) p_n \\ &= \sum_{n=0}^{\infty} F_X^{*n}(x) p_n \end{aligned} \tag{2}$$

其中，$p_n = P(N = n)$是频率分布，$F_X(x)$是强度分布，$F_X^{*n}(x)$是强度分布的 n 重卷积。如果随机变量X仅在 0，1，2…取值，那么 n 重卷积的计算方法如下：

$$F_X^{*n}(x) = \begin{cases} I(x \geq 0) & n = 0 \\ F_X(x) & n = 1 \\ \sum_{y=0}^{x} F_X^{*(n-1)}(x-y) f_X(y) & n = 2, 3, \ldots \end{cases} \tag{3}$$

3.1 连续分布的离散化

为什么要对连续分布进行离散化？通常我们假设索赔强度分布是连续分布，虽然理论上可以这么讲，但是实际操作中通常会采用一些数值计算方法来计算复合分布，这些方法要求索赔强度具有离散的分布，因此需要对现有的连续分布进行离散化处理。在某种程度上，离散化更加接近实际，比如损失额通常是整数倍的货币单位。

所谓离散化就是将连续分布的支集区域划分为若干小区域，然后以这个区域中的某一个点代替原来连续分布在这片区域的取值概率。这个"代表点"可以是这个区域的左右端点，也可以是区域中点。此外，通常只对分布的"主体"进行离散化，什么叫做分布的"主体"？以正态分布为例，其分布的支集为$(-\infty, \infty)$，显然不可能对其所有取值范围进行离散化，由于正态分布在两侧的取值概率很小，可以忽略不计，我们于是可以以均值为中心，以若干倍标准差为半径划定一个区域，在这个区域上进行离散化，这个区域上的分布函数就是该分布

的“主体”，区域的大小则依赖于研究的精确程度。定义$F(x)$的为连续分布函数，f_x为离散化后的概率函数。目前，actuar 包中的 discretize 函数支持四种离散化方法。

1）上端离散化，或者说对$F(x)$向前微分。

$$f_x = F(x+h) - F(x) \tag{4}$$

对于$x = a, a+h, \ldots, b-h$，离散化后的 cdf 总是在原 cdf 之上。

2）下端离散化，或者说对$F(x)$向后微分。

$$f_x = \begin{cases} F(a) & x = a \\ F(x) - F(x-h) & x = a+h, \ldots, b \end{cases} \tag{5}$$

离散化后的 cdf 总是在原 cdf 之下。

3）中点离散化。

$$f_x = \begin{cases} F(a+h/2) & x = a \\ F(x+h/2) - F(x-h/2) & x = a+h, \ldots, b-h \end{cases} \tag{6}$$

原 cdf 正好从中间穿过离散化后的 cdf。

4）无偏离散化，或者说是一阶矩局部匹配法。

$$f_x = \begin{cases} \frac{E(X\wedge a) - E(X\wedge a+h)}{h} + 1 - F(a) & x = a \\ \frac{2E(X\wedge x) - E(X\wedge x-h) - E(X\wedge x+h)}{h} & a < x < b \\ \frac{E(X\wedge b) - E[X\wedge b-h]}{h} - 1 + F(b) & x = b \end{cases} \tag{7}$$

离散后的分布和原分布在区间$[a, b]$内有相同的取值概率和期望值。

discretize 函数返回的是一串f_x概率值，如果要对其进行做图需要进行特殊处理，其语法如下：

```
discretize(cdf, from, to, step = 1,
          method = c("upper", "lower", "rounding",
"unbiased"),
          lev, by = step, xlim = NULL)
```

注意事项：

1）cdf 必须是一个含 x 的表达式。

2）from 和 to 分别指定 a 和 b，也就是分布主体的范围，step 指定 h。

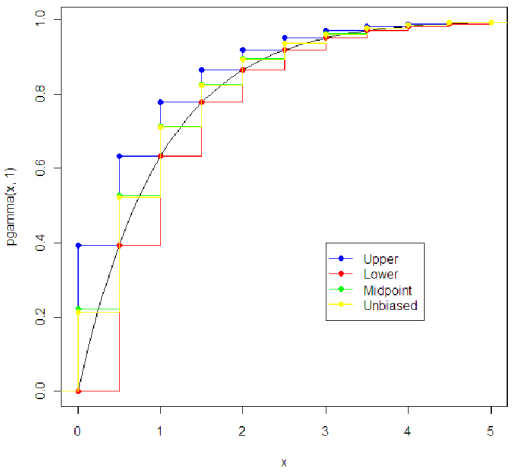
3）lev 只在 method=“unbiased”时才指定。

4）by 和 xlim 与前面参数等价，详见帮助文档。

例子：

```
#假设要对Gmmma(1,1)，进行离散化
#上端和下端离散化
> fu = discretize(pgamma(x, 1), method = "upper",
+      from = 0, to = 5, step = 0.5)
> fl = discretize(pgamma(x, 1), method = "lower",
+      from = 0, to = 5, step = 0.5)
> curve(pgamma(x, 1), xlim = c(0, 5))
#给定x的序列，用来做阶梯函数图像。
```

```
> x =seq(0, 5, 0.5)
> plot(stepfun(head(x, -1), diffinv(fu)), pch = 19,
+      ,col='blue',add = TRUE)
> plot(stepfun(x, diffinv(fl)), pch = 19,
+      ,col='red',add = TRUE)
# 中点离散化
> fr = discretize(pgamma(x, 1), method = "rounding",
+      from = 0, to = 5, step = 0.5)
> plot(stepfun(head(x, -1), diffinv(fr)), pch = 19,
+      ,col='green',add = TRUE)
# 无偏离散化
> fb = discretize(pgamma(x, 1), method = "unbiased",
+      lev = levgamma(x, 1), from = 0, to = 5, step = 0.5)
> plot(stepfun(x, diffinv(fb)), pch = 19,
+      ,col='yellow',add = TRUE)
> legend(3,0.4,legend=c("Upper","Lower","Midpoint","Unbiased")
+        ,col=c('blue','red','green','yellow'),pch=19,lty=1)
```



3.2 复合分布的计算

当我们对索赔强度分布离散化后，我们可以使用一些算法计算复合分布。函数 aggregateDist 提供了五种方法，下面仅进行简单介绍，对细节感兴趣的读者可以参考帮助文档：

1）递推法。频率分布仅支持$(a,b,0)$分布族和$(a,b,1)$分布族，也就是泊松，二项，几何和负二项分布及其修正分布，强度分布需要输入离散后的分布。

2）卷积法。使用(2)和(3)进行计算，频率分布可以是任何离散分布，强度分布需要输入离散化后的分布。这种方法只能解决小型问题。

3）正态近似法。给定频率分布和强度分布就可以利用公式

$$\mu_S = E(S) = E(N) * E(X) \tag{8}$$

$$\sigma_S^2 = Var(S) = E(N)Var(X) + Var(N)E(X)^2 \tag{9}$$

计算复合分布的均值和方差，再利用

$$F_S(x) \approx \Phi(\frac{x-\mu_S}{\sigma_S}) \tag{10}$$

计算复合分布的分布函数。

4）正态幂近似法（Normal Power approximation）

$$F_S(x) = \Phi(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x-\mu_S}{\sigma_S}}) \tag{11}$$

其中γ_S是偏度系数。正态幂近似法的原理是对标准化后的随机变量S泰勒展开为标准正态随机变量及其 2 次幂的线性组合，也就是另$S \approx g(Y)$，Y服从标准正态分布。这种近似法在$x > \mu_S$且$\gamma_S < 1$时是可以进行的，也就是可以对分布的右尾进行拟合。

1）随机模拟法。从S中随机抽样，再用S的经验分布近似$F_S(x)$。这种方法可以调用 simul 函数（后面要讲）对频率分布和强度分布进行模拟，适用于复杂系统的建模。

aggregateDist 的参数依据所选方法的不同而不同。从下面函数的语法可以看出，该函数是先定方法，再选参数，因此在用法说明中，只介绍和此种方法有关的参数。

函数语法：

```
aggregateDist(method = c("recursive", "convolution", "normal",
                         "npower", "simulation"),
              model.freq = NULL, model.sev = NULL, p0 = NULL,
              x.scale = 1, moments, nb.simul, ...,
              tol = 1e-06, maxit = 500, echo = FALSE)
```

用法说明：

1）递推法中：method="recursive"，model.freq 必须是 "binomial", "geometric", "negative binomial", "poisson"中的一种，分布参数可以在省略号处指定，但是参数名称必须要与这四个分布规定的一致。model.sev 是一个向量，向量的每个元素依次是X取 0，1，2...个货币单位的概率，注意这个向量的第一个元素必须是X取 0 的概率，如果X不能取到某个值（比如 2），那么向量的对应位置（这里是第三个元素）取 0。通常 model.sev 可以直接使用 discretize 的结果。P0 是频数分布在$N=0$时的概率[1]。， x.scale 指定X的货币单位，比如 1 元，100 元等。

2）卷积法中：method="convolution"，model.freq 是一个向量，向量的每个元素依次是N取 0，1，2...的概率，第一个元素必须是N取 0 的概率。model.sev 的使用方法与递推法相同。x.scale 指定X的货币单位，比如 1 元，100 元等。

3）正态近似法和正态幂近似法：method="normal"或"npower"，moments 是一个向量，moments=c(μ_S，σ_S^2，γ_S)，正态近似只需指定前两个元素即可。注意这种近似法需要满足$x > \mu_S$且$\gamma_S < 1$，否则会发生报错。

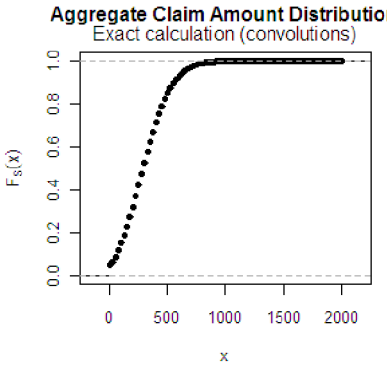
4）模拟法中：method="simulation"，具体使用方法可以参考后文中对 simul 函数的介绍。nb.simul 是模拟的次数。

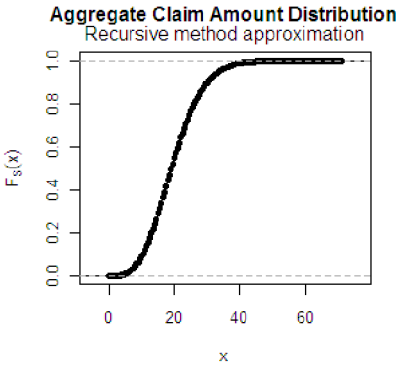
函数返回的是一个 aggregateDist 对象，可以对其进行五数总括（summary），输出结果（print），求均值（mean），求分位数（quantile），做图（plot），求节点（konts）

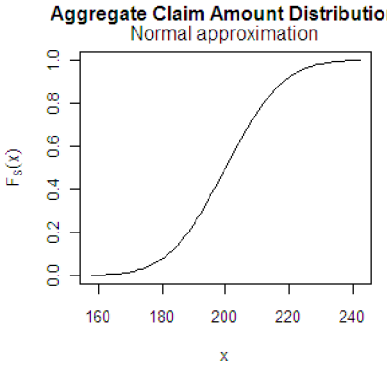
[1]也许你会问$(a,b,0)$分布族四种分布的$P(N=0)$不是已经确定好的吗？可是有时，这个概率需要进行修正，比如某个风险集合出现 0 次赔付的概率很大，而我们又认为出现 1 次以上赔付的概率分布形状类似于 poisson 分布，那么就需要设$p_0 = 0.8$（比方说），之后的$p_1, p_2, \ldots$共同划分剩余的 0.2 的概率。调整方法很简单，调整后的分布概率=调整前的分布概率×调整系数即可。我们把调整后的的分布族称作$(a,b,1)$分布族。

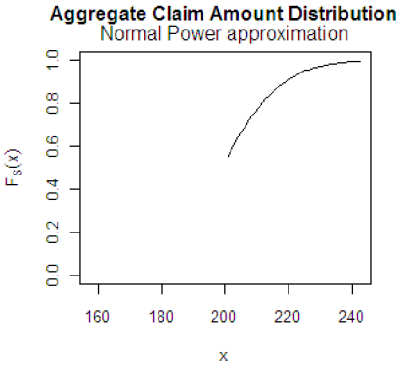
等操作。
例子：

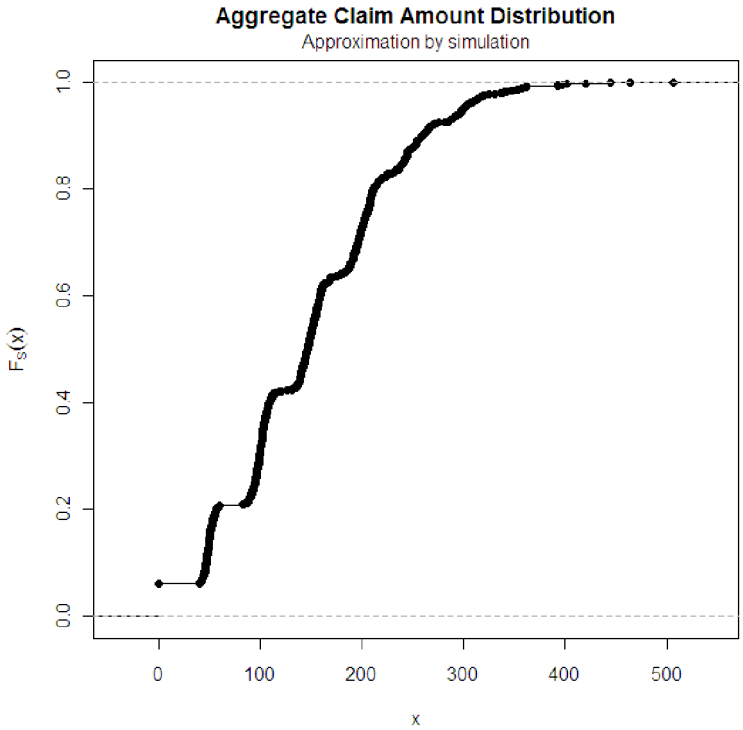
```
> par(mfrow=c(2,2))
#卷积法，强度分布，货币单位数X从0到10
> fx1 = c(0, 0.15, 0.2, 0.25, 0.125, 0.075,
+      0.05, 0.05, 0.05, 0.025, 0.025)
#频数分布，N从0到8
> pn1 = c(0.05, 0.1, 0.15, 0.2, 0.25, 0.15, 0.06, 0.03, 0.01)
#货币单位设为25
> Fs1 = aggregateDist("convolution", model.freq = pn1,
+      model.sev = fx1, x.scale = 25)
#看看最大值是不是10×25×8=2000?
> plot(Fs1)
#递推法，首先对Gamma分布进行离散化
> fx2=discretize(pgamma(x,2,1), from = 0, to = 22,
+      step=0.5, method = "unbiased", lev=levgamma(x,2,1))
#频数分布选用poisson分布,特别指定poisson分布参数lambda=10
> Fs2=aggregateDist("recursive", model.freq = "poisson",
+      model.sev = fx2, lambda = 10, x.scale = 0.5)
> plot(Fs2)
#正态近似和正态幂近似，注意正态幂近似的有效范围。
> Fs3=aggregateDist("normal",moments=c(200,200))
> plot(Fs3)
> Fs4=aggregateDist("npower",moments=c(200, 200, 0.5))
> plot(Fs4)
#模拟法，可以留到后面再看。
> model.freq=expression(data = rpois(3))
> model.sev=expression(data = rgamma(100, 2))
> Fs5=aggregateDist("simulation", nb.simul = 1000,
+      model.freq, model.sev)
> mean(Fs5)
[1] 148.1398
> summary(Fs5)
Aggregate Claim Amount Empirical CDF:
    Min.  1st Qu.   Median    Mean  3rd Qu.    Max.
  0.0000  94.9694 146.4297 148.1398 203.7051 506.7553
> quantile(Fs5,0.5)
     50%
146.4297
> plot(Fs5)
```


Aggregate Claim Amount Distribution
Exact calculation (convolutions)
$F_S(x)$
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
x


Aggregate Claim Amount Distribution
Recursive method approximation
$F_S(x)$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
x


Aggregate Claim Amount Distribution
Normal approximation
$F_S(x)$
0.0
0.2
0.4
0.6
0.8
1.0
160
180
200
220
240
x


Aggregate Claim Amount Distribution
Normal Power approximation
$F_S(x)$
0.0
0.2
0.4
0.6
0.8
1.0
160
180
200
220
240
x


Aggregate Claim Amount Distribution
Approximation by simulation
$F_S(x)$
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500
x