用 R 也能做精算—actuar 包学习笔记（二）

李皞
（中国人民大学 统计学院 风险管理与精算）

2.损失分布

2.1 损失分布种类

根据损失额的特征，损失分布常选用具有非负支集（密度函数$f(x)$的支集指的是使得$f(x) \neq 0$的 x 的集合）的连续分布。R 中对于一些分布提供了 d，p，q，r 四种函数，分别是密度函数、分布函数、分布函数的反函数（分位数）和生成该分布的随机数。actuar 包提供了与 Loss Models（Klugman 等，2009）的附录 A 中所列示的连续分布族相配套的这四种函数（除去逆高斯和对数 t 分布，但包括对数 Gamma 分布），这些分布中 R 的基础包 stats 中并不自带，但有些分布在精算研究中却很重要（比如 pareto 分布）。此外，actuar 包还对这些连续分布提供了 m、lev 和 mgf 三种函数，m 是计算理论原点矩，lev 是计算有限期望值，mgf 是计算矩母函数。密度函数、分布函数、原点矩、有限期望值及其 k 次方都可以通过查表得到[1]。

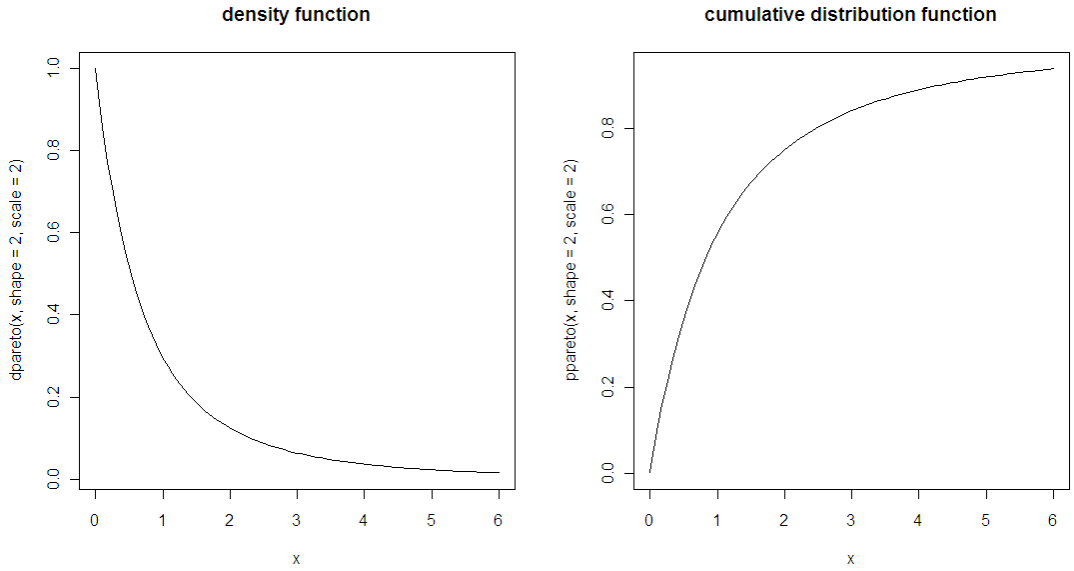
对于经验数据，如上面介绍 actuar 包中提供了 emm 和 elev 来计算经验原点矩和经验有限期望值（这两个函数的前缀都是 empirical）。

需要注意的是，这些分布有的需要指定 rate 参数或 scale 参数，scale=1/rate，因此两者在本质上是等价的，Loss Models 的附录 A 中使用的是 scale 参数，在指定参数时千万不要弄混。

例子：

```
#这里以双参数 pareto 分布为例
> par(mfrow=c(1,2))
#绘制密度函数曲线
> curve(dpareto(x, shape=2,scale=2),from=0.001,to=6,
main="density function")
#绘制分布函数曲线
> curve(ppareto(x, shape=2, scale=2),from=0.001,to=6,
main="cumulative distribution function")
```

[1] http://www.soa.org/files/pdf/edu-2009-fall-exam-c-table.pdf



#求pareto分布中位数

```
> qpareto(0.5, shape=2, scale=2)
[1] 0.8284271
```

#生成5个pareto分布随机数

```
> rpareto(5, shape=2, scale=2)
[1] 1.4223168 0.1153321 0.1013023 2.7571208 1.8178389
```

#求$E(X^{1.5})$，注意pareto分布k阶矩要求$-1 < k < \alpha$，order=k，α是shape参数

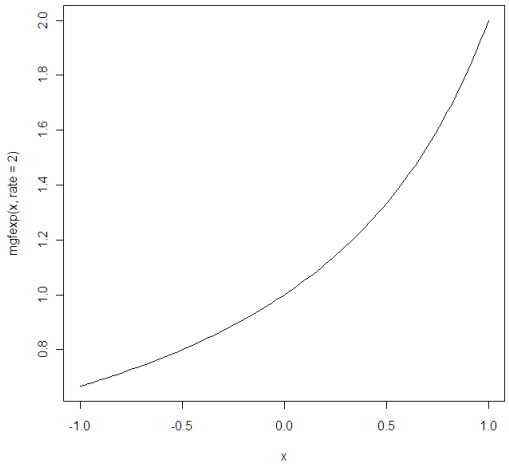
```
> mpareto(order=1.5, shape=2, scale=2)
```

#求$E[(X \wedge 5)^{1.5}]$，注意同样要求$-1 < k < \alpha$，order=k，α是shape参数

```
> levpareto(limit=5, shape=2, scale=2, order = 1.5)
[1] 2.355089
```

#指数分布矩母函数，均值=1/2。矩母函数形式为$M_X(t) = \mu/(\mu - t)$，μ为rate参数。

```
> curve(mgfexp(x, rate = 2),-1,1)
```



2.2 损失分布的估计

矩估计和极大似然估计是分布参数估计的基本方法。在 R 中，MASS 包中的 fitdistr

函数可以进行极大似然估计。在 actuar 包中，mde 函数则提供了三种基于距离最小化的分布拟合方法（minium distance estimates）。

1）Cramér-von Mises 方法（CvM）最小化理论分布函数和经验分布函数（对于分组数据是 ogive）的距离。

未分组数据：

$$d(\theta) = \sum_{j=1}^{r} w_j (F(x_j;\theta) - F_n(x_j))^2 \tag{1}$$

分组数据：

$$d(\theta) = \sum_{j=1}^{r} w_j (F(c_j;\theta) - \tilde{F}_n(c_j))^2 \tag{2}$$

在这里，$F(x;\theta)$是理论分布函数，θ是其参数；$F_n(x)$是经验分布函数 ecdf；$\tilde{F}_n(x)$是分组数据的经验分布函数 ogive；w_j是赋予每个观测或组别的权重，默认都取 1。

2）修正卡方法仅应用于分组数据，通过最小化各组期望频数与实际观测频数的平方误差得到。

$$d(\theta) = \sum_{j=1}^{r} w_j [n(F(c_j;\theta) - F(c_{j-1};\theta)) - n_j]^2 \tag{3}$$

其中，$n = \sum_{j=1}^{n} n_j$，w_j默认情况下为n_j^{-1}。

3）LAS 法（layer average severity）也仅应用于分组数据。通过最小化各组内的理论和经验有限期望函数的平方误差得到。

$$d(\theta) = \sum_{j=1}^{r} w_j (LAS(c_{j-1}, c_j;\theta) - L\tilde{A}S_n(c_{j-1}, c_j))^2 \tag{4}$$

其中$LAS(x,y) = E(X \wedge y) - E(X \wedge x)$，$L\tilde{A}S_n(x,y) = \tilde{E}_n[X \wedge y] - \tilde{E}_n[X \wedge x]$，$E()$是理论分布的有限期望函数，而$\tilde{E}_n$是经验分布的有限期望函数。$w_j$默认情况下为$n_j^{-1}$。

函数调用 stats 包中的 optim 函数做最优化，其语法为：

```
mde(x, fun, start, measure = c("CvM", "chi-square", "LAS"),
    weights = NULL, ...)
```

用法说明：

1）x 是分组数据对象的或未分组的数据。

2）fun 是待拟合的分布，CvM 法和修正卡方法需要指定分布函数：p**。LAS 法需要指定理论有限期望函数 lev**。

3）start 指定参数初始值。形式必须以列表的形式，形式可以见例子，有几个参数就要指定几个初始值。

4）measure 是指定方法。weight 指定权重，否则采用默认权重。

5）…是其他参数，可以指定 optim 函数中的参数，比如使用 L-BFGS-B 方法进行优化可以添加参数 method="L-BFGS-B"。

我们可以对上面的 gdental 数据进行分布拟合，这是一个分组数据。

例子：

#假设数据来自均值为 200 的指数分布

```
> hist(gdental)
```
#mde输出的结果是一个列表(list)，rate是参数估计结果，distance是最小化后的距离。

#CvM法

```
> (mde.est1=mde(gdental,pexp,start=list(rate=1/200),measure="CvM"))
     rate
 0.003551270
   distance
 0.002841739
> mu1=mde.est1[[1]]
```

#修正卡方法

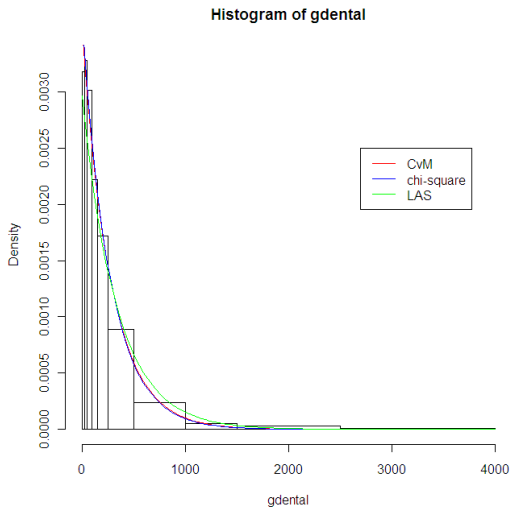
```
> (mde.est2=mde(gdental,pexp,start=list(rate=1/200),measure="chi-square"))
     rate
 0.003639893
   distance
  13.54132
> mu2=mde.est2[[1]]
```

#LAS法

```
> (mde.est3=mde(gdental,levexp,start=list(rate=1/200),measure="LAS"))
> mu3=mde.est3[[1]]
     rate
 0.002966064
  distance
  694.5385
```

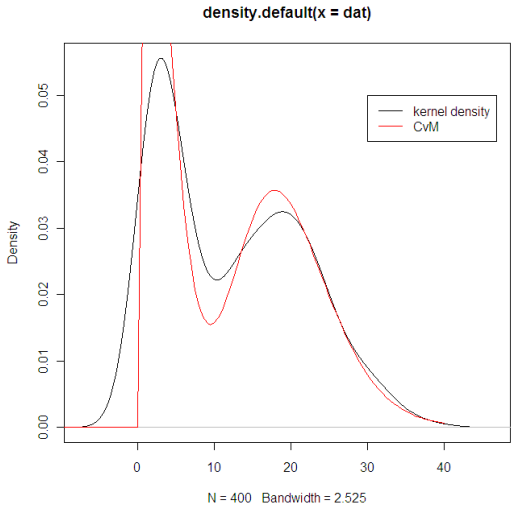
#做图，可以看出CvM法和修正卡方法估计结果相似，而LAS法损失分布尾部拟合效果较好。

```
> curve(mu1*exp(1)^(-mu1*x),from=0,to=4000,add=T,col='red')
> curve(mu2*exp(1)^(-mu2*x),from=0,to=4000,add=T,col='blue')
> curve(mu3*exp(1)^(-mu3*x),from=0,to=4000,add=T,col='green')
> legend(2700,0.0025,legend=c("CvM","chi-square","LAS"),
    +   col=c('red','blue','green'),lty=1)
```



我们还可以对非分组数据进行分布拟合，下面的一个例子是一个混合分布。

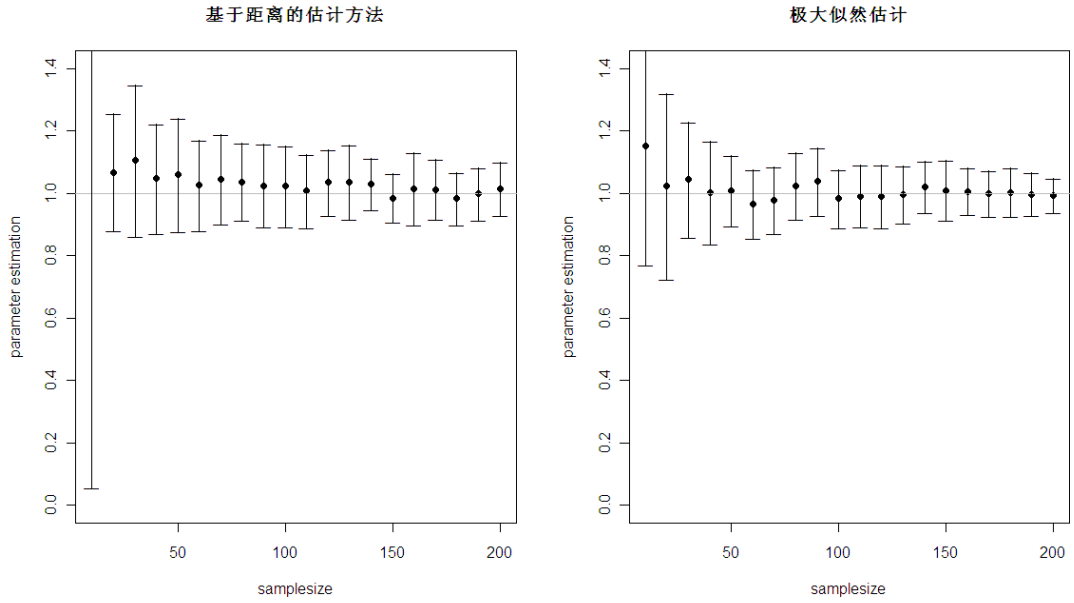
```
#生成400个随机数
#其中200个来自Gamma(α = 2, θ = 2)，200个来自Gamma(α = 10, θ = 2)
>dat=c(rgamma(200,shape=2,scale=2),rgamma(200,shape=10,scale=2
))
#混合分布密度
> dfn=function(x,a,alpha1,alpha2,theta)
+ {
+   a*dgamma(x,shape=alpha1,scale=theta)+
+          (1-a)*dgamma(x,shape=alpha2,scale=theta)
+ }
#混合分布函数
> pfn=function(x,a,alpha1,alpha2,theta)
+ {
+   a*pgamma(x,shape=alpha1,scale=theta)+
+     (1-a)*pgamma(x,shape=alpha2,scale=theta)
+ }
#mde估计，对于非分组数据只能使用CvM法
> mde.est4=mde(dat,pfn,start=list(a=0.4,alpha1=1,alpha2=8,
+   theta=2.5),measure="CvM")
#参数输出
> (para=mde.est4[[1]])
        a     alpha1     alpha2      theta
 0.4765863  1.9676536 10.4741994  1.8946438
> plot(density(dat))
> curve(dfn(x,a=para[1],alpha1=para[2],alpha2=para[3],theta=para[4]),
+       from=-10,to=40,col='red',add=T)
> legend(30,0.05,legend=c('kernel density','CvM'),
+       col=c("black",'red'),lty=1)
```



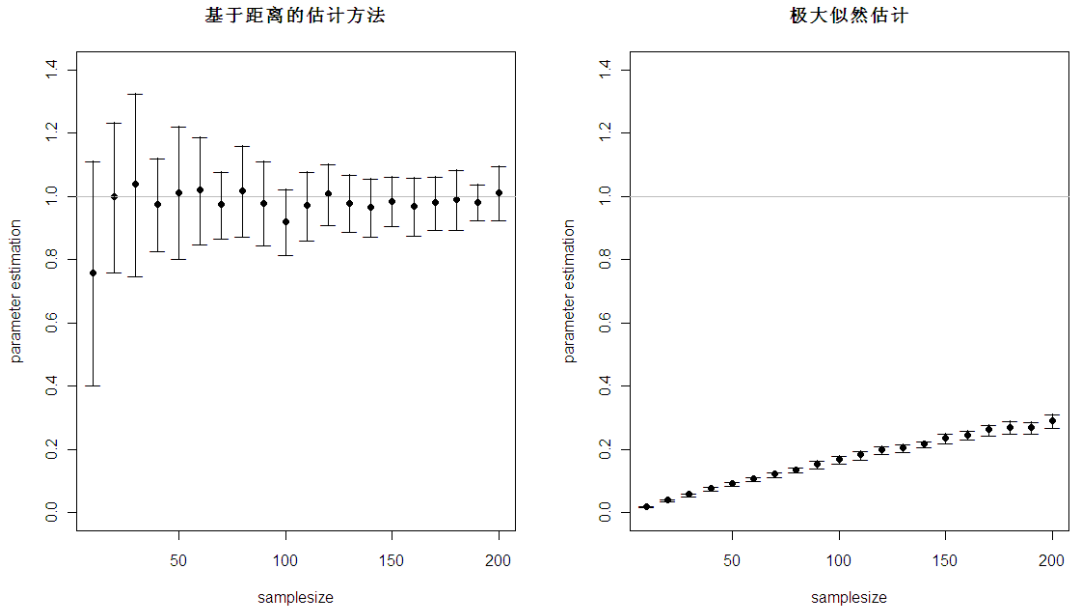
在此，我们感兴趣的是将基于距离的分布拟合方法与极大似然估计的参数估计效果进行以下对比。因此不妨做一个实验：

简单起见，先从单参数拟合问题开始，这是一个一维优化问题。

首先生成 50 组来自于 rate=1 的指数分布随机数，每组的个数都为 10。然后，对于每一组随机数，分别用基于距离的估计方法和极大似然估计进行参数估计，将 50 次模拟结果的均值和标准差记录下来。之后，将随机数的个数由 10 增加到 20，30…200。加大样本量，重复之前的过程。最终得到的结果如下图：



可以看出，在单参数估计中，尤其是在样本量较大时，两种方法估计的结果相差不大，而极大似然估计的方差要比基于距离的估计方差略小。那么，两种方法对于异常值的稳健性如何呢？在上面生成的所有组随机数中，先剔除两个指数分布随机数，再混入两个来自[200,300]均匀分布的随机数，再重新对参数进行估计，结果很明显，基于距离的估计方法估计的结果很稳定，而极大似然估计的参数结果受异常值的影响很大！因此，对于经常存在异常值的损失数据，使用基于距离的分布拟合方法往往更加稳健。



对于两参数的估计，由于某些分布要求参数恒为正，使用 mde 函数经常会报错，通常的解决办法是估计参数的对数形式：

$$\theta = \exp(\tau) \tag{5}$$

当算法在迭代时，让τ在$(-\infty, \infty)$范围内变动，同时θ恒为正。当估计出τ的值后，再取指数进行还原。这样可以有效的降低优化失败的次数。

例子：

```
> pgammalog=function(x,logshape,logscale)
+ {
+   pgamma(x,exp(logshape),exp(logscale))
+ }
>
> aa=rgamma(200,shape=3,scale=1)
>
estlog=mde(aa,pgammalog,start=list(logshape=1.3,logscale=0.2),
measure='CvM',method='L-BFGS-B',lower=c(0.5,-0.5),upper=c(1,5,
0.5))$estimate
> exp(estlog)
 logshape  logscale
2.7182818 0.9641508
```

2.3 损失随机变量的修正

我们知道，由于某些保险条款规则的存在，保险实际赔付额往往和实际损失数额是不相等的。假定我们定义实际损失数额为随机变量X，实际赔付额为Y，那么$Y = f(X)$。实际赔付额这个f究竟是什么，具体可以分为以下几个因素：

- 免赔额（deductible）：损失额超过某个数额，则赔付超出的部分。具体可以分为一般免陪额（ordinary deductible）和绝对免赔额（franchise deductible）。

  一般免赔额的数学形式：

$$Y = (X-d)_+ = \begin{cases} 0 & X \le d \\ X-d & X > d \end{cases} \tag{6}$$

  绝对免赔额的数学形式：

$$Y = \begin{cases} 0 & X \le d \\ X & X > d \end{cases} \tag{7}$$

- 最大保障损失（maximum covered loss）：是保险人对于单个损失支付的最大赔付额。无论是否有免赔额，只要损失额超过最大保障损失就以最大保障损失封顶。

  数学形式：

$$Y = X \wedge u = \begin{cases} X & X \le u \\ u & X > u \end{cases} \tag{8}$$

  其中随机变量$X \wedge u$称为有限损失随机变量，其期望$E(X \wedge u)$称为有限期望值（也就是

前面介绍的 lev）。前面提到过保单限额的概念，在那里也用符号u表示。保单限额（policy limit）与最大保障损失的区别在于：当存在免赔额时，保单限额=最大保障损失-免赔额。因此当不存在免赔额时，两者是等价的。

- 通货膨胀和共同保险（coinsurance）：通货膨胀是指未来的赔付额等于当前损失额乘以一个通胀因子，$Y=(1+r)X$；而共同保险是指对每一次损失，保险公司只赔付一定的比例$Y=\alpha X(0<\alpha<1)$，其余部分则由投保人自行承担。之所以将这两者放在一起，是因为它们都是对原始损失变量乘以一个常量得到赔付额，在没有免赔额和赔偿限额的情况下，两者形式上是相同的。当存在赔额和赔偿限额时，通货膨胀的数学形式为：

$$Y=\begin{cases}0 & (1+r)X\le d\\ (1+r)X-d & d<(1+r)X\le u\\ u-d & (1+r)X>u\end{cases} \tag{9}$$

而共同保险的数学形式为：

$$Y=\begin{cases}0 & X\le d\\ \alpha(X-d) & d<X\le u\\ \alpha(u-d) & u<X\end{cases} \tag{10}$$

可以看出，通货膨胀首先对随机变量X进行通胀修正，再进行免赔额和赔偿限额的修正；共同保险首先对随机变量X进行免赔额和赔偿限额的修正，再进行共保修正。两者的顺序是不同的。

严格的说，Y是可以进一步区分为 cost per loss（Y_L）和 cost per payment（Y_P）。两者的区别在于，Y_L是指每次损失带来保险公司的赔付额，而Y_P是指每次赔付带来保险公司的赔付额。如果没有免赔额，那么两者自然是等价的，但是由于免赔额的存在，每次损失保险公司不一定要赔偿，Y_L可以等于 0，而每次赔付保险公司必然有正的赔付额，Y_P严格大于 0。因此当损失额X大于免赔额，$Y_P=Y_L$，否则$Y_L=0$而Y_P没有定义。以一般免赔额为例：

$$Y_L=(X-d)_+=\begin{cases}0 & X\le d\\ X-d & X>d\end{cases} \tag{11}$$

$$Y_P=\begin{cases}\text{无定义} & X\le d\\ X-d & X>d\end{cases} \tag{12}$$

为什么会有Y_L和Y_P的区分？通常保险公司只能获得实际赔付额的数据，由于免赔额的存在，当损失额小于免赔额d时，被保险人可能根本不会去保险公司报案，因此小于d的损失数据要么不全，要么干脆无法获得，因此精算师索性将这部分损失数据忽略，只考虑导致正的赔付额的损失。关键就在这里！如果只考虑正的赔付额那么Y_P就是一个条件随机变量，也就是以$X>d$为条件，也就是：

$$Y_P=Y_L|X>d \tag{13}$$

这样Y_P的分布就是一个条件分布，而且Y_P的取值恒大于 0。

当精算师需要根据赔付数据Y对损失额X进行建模时，就需要建立两者之间的联系。什么样的联系呢？就是根据两个随机变量间的变换关系得到两个随机变量间分布密度和分布

函数间的关系。大体的建模过程可以表述为：

(1) 假定X的原始分布，然后根据变换关系得到Y的修正分布。在这个过程中，原始分布的参数也同时传递到了修正分布中。

(2) 将实际赔付数据带入修正分布中，使用极大似然或其他估计方法估计得到修正分布的参数。

(3) 使用逆变换法得到原始分布的参数。

actuar 包中，coverage 这个函数可以完成将原始分布变换为修正分布的工作。coverage()输出的是一个函数对象，其语法如下：

```
coverage(pdf, cdf, deductible = 0, franchise = FALSE,limit
       = Inf, coinsurance = 1, inflation = 0,per.loss = FALSE)
```

使用提示：

1）如果 pdf 和 cdf 同时指定，那么输出是修正后的 pdf，如果只指定 cdf，那么输出的是修正后的 cdf。特别注意的是如果存在 deductible 或 limit，那么 cdf 必须指定。

2）deductible 设置免赔额d，franchise 控制是绝对免赔额还是一般免赔额，默认为 FALSE，即一般免赔额。

3）limit 设置最大保障损失u，默认为无上限。

4）coinsurance 是共保因子α，取值为 0-1 之间的数。

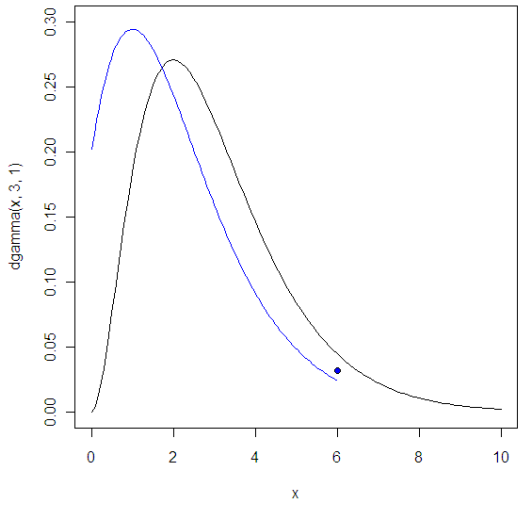
5）inflation 是通胀率r，取值为 0-1 之间的数。

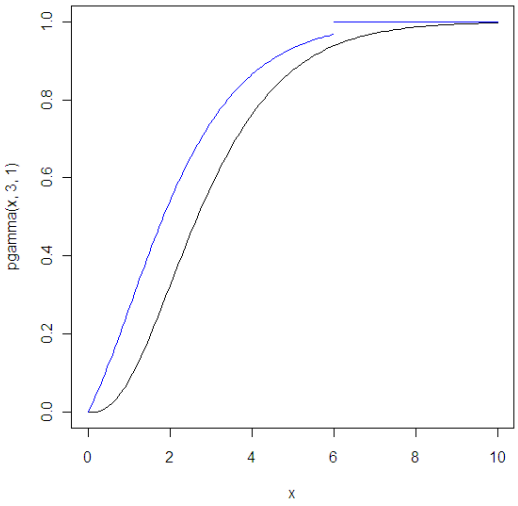
6）per.loss 控制是采用Y_P还是Y_L，默认采用Y_P。

例子：

假设原始损失服从形状参数 shape=3，尺度参数 scale=1 的 Gamma 分布。

```
#修正后的密度函数
> f = coverage(dgamma, pgamma, deductible = 1, limit = 7)
#修正后的密度函数与原始密度函数的比较，由于修正后的密度函数是连续分布与离散分布的混合，因此probability mass用加粗的点标出。
> curve(dgamma(x, 3, 1), xlim = c(0, 10), ylim = c(0, 0.3))
> curve(f(x, 3, 1), xlim = c(0.01, 5.99), col = 4, add = TRUE)
> points(6, f(6, 3, 1), pch = 21, bg = 4)
#修正后的分布函数
> F = coverage(cdf = pgamma, deductible = 1, limit = 7)
#修正后的密度函数与原始密度函数的比较，probability mass处cdf出现跳跃。
> curve(pgamma(x, 3, 1), xlim = c(0, 10), ylim = c(0, 1))
> curve(F(x, 3, 1), xlim = c(0, 5.99), col = 4, add = TRUE)
> curve(F(x, 3, 1), xlim = c(6, 10), col = 4, add = TRUE)
```


dgamma(x, 3, 1)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
2
4
6
8
10
x


pgamma(x, 3, 1)
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
x