

Loss modifications

Milica Cudina

2023-02-06

The model

I am choosing the exponential model for my demonstration. More precisely, my loss random variable will be

$$X \sim \text{Exponential}(\text{mean} = \theta = 100)$$

Let me set the value of the parameter of my exponential:

```
theta=100
```

Simulated losses

I choose to simulate 10000 losses. So, this is how I define my number of draws from the above exponential:

```
nsim=10000
```

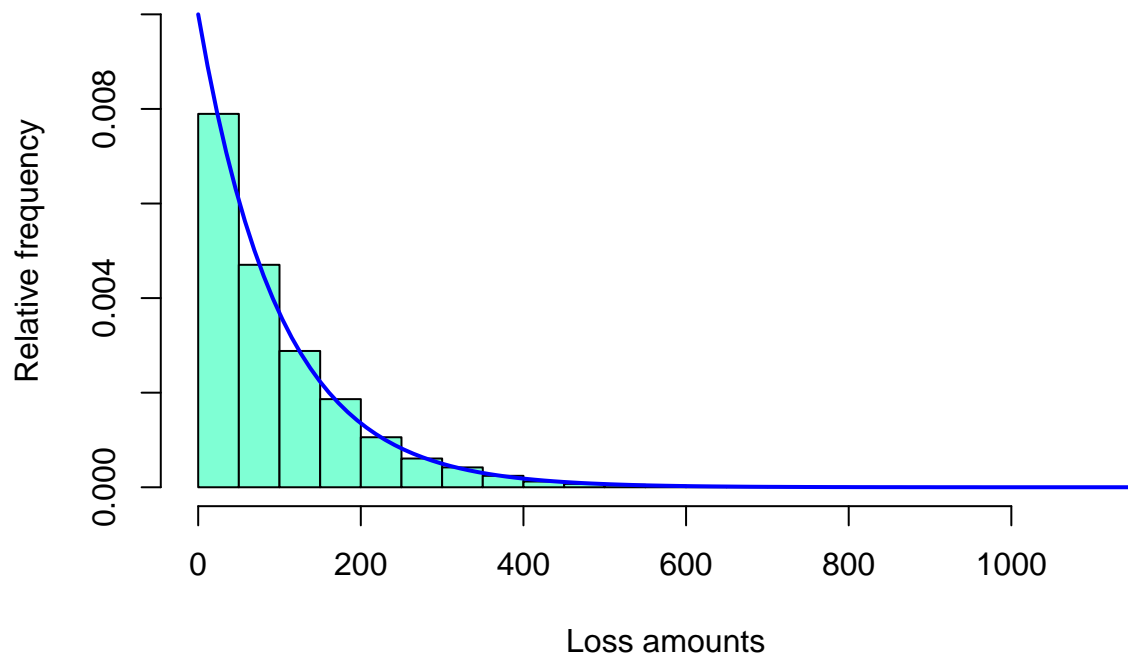
Here is how I obtain my vector of draws from the exponential:

```
sims=rexp(nsim, rate=1/theta)
```

Let's plot the histogram of the simulated values, along with the theoretical density.

```
hist(sims, breaks=25, ylim=c(0, dexp(0, rate=1/theta)),  
      main="Simulated losses",  
      xlab="Loss amounts",  
      ylab="Relative frequency",  
      prob=TRUE, col="aquamarine")  
curve(dexp(x, rate=1/theta), col="blue", lwd=2, add=TRUE)
```

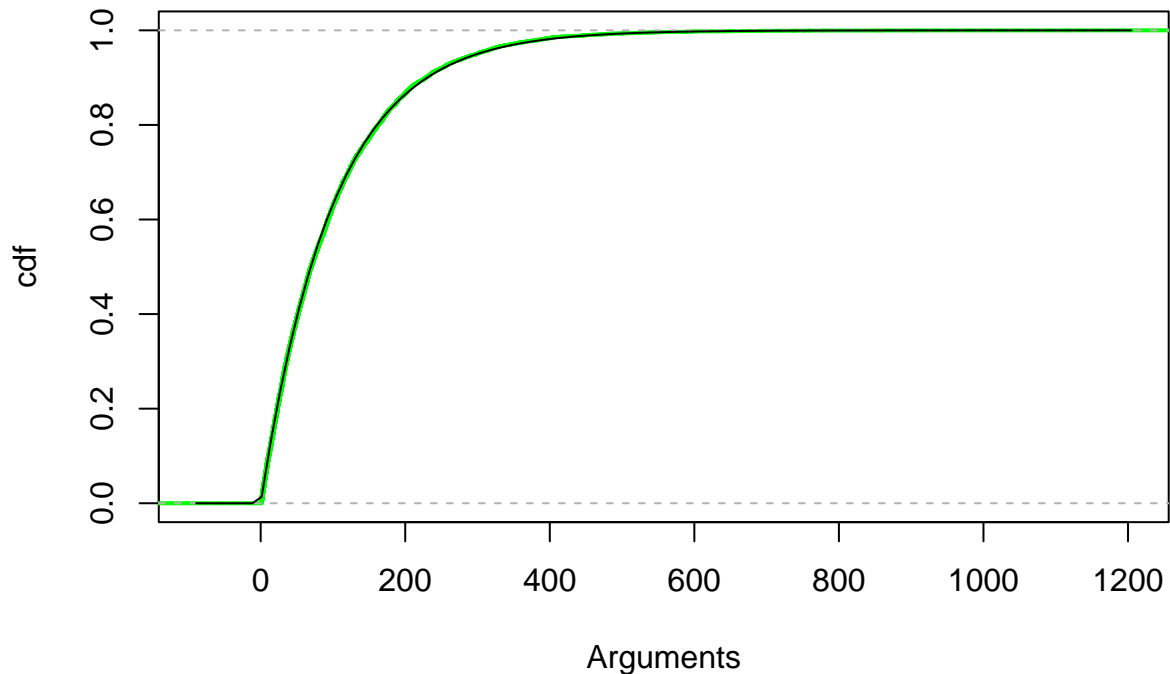
Simulated losses



We can create a similar plot for the empirical cumulative distribution function. This is the function which counts how many values in your data set were less than or equal to its argument. There is a command in R which constructs the empirical cumulative distribution function. Enter `?ecdf` into your console to learn more about it. I will also add the theoretical plot of the cumulative distribution function. Do the two match well?

```
cum.sims=ecdf(sims)
plot(cum.sims,
     main="Empirical cdf",
     xlab="Arguments",
     ylab="cdf",
     col="green", lwd=2)
curve(pexp(x, rate=1/theta), add=TRUE)
```

Empirical cdf



The ordinary deductible

Let us say that the above losses are insured using an insurance policy with an ordinary deductible of 50. Let's set its value up in R:

```
d=50
```

You might ask yourself what the probability is that the deductible is met. Here is how you get it in R:

```
p.d=1-pexp(d, rate=1/theta)
p.d
```

```
## [1] 0.6065307
```

Here is how you get the proportion of the losses which met the deductible in the above batch of simulated values.

```
ind.d=(sims>d)
prop.d=mean(ind.d)
prop.d
```

```
## [1] 0.6052
```

What do you notice?

The per-payment random variable

In R is pretty easy to “condition” on the deductible being met. You simply just keep the quantities among the losses which exceed the deductible and subtract d .

```
y.p=sims[sims>d]-d
```

Here is the number of remaining amounts that the insurer has a liability for.

```
length(y.p)
```

```
## [1] 6052
```

Here is the empirical mean of the per payment random variable.

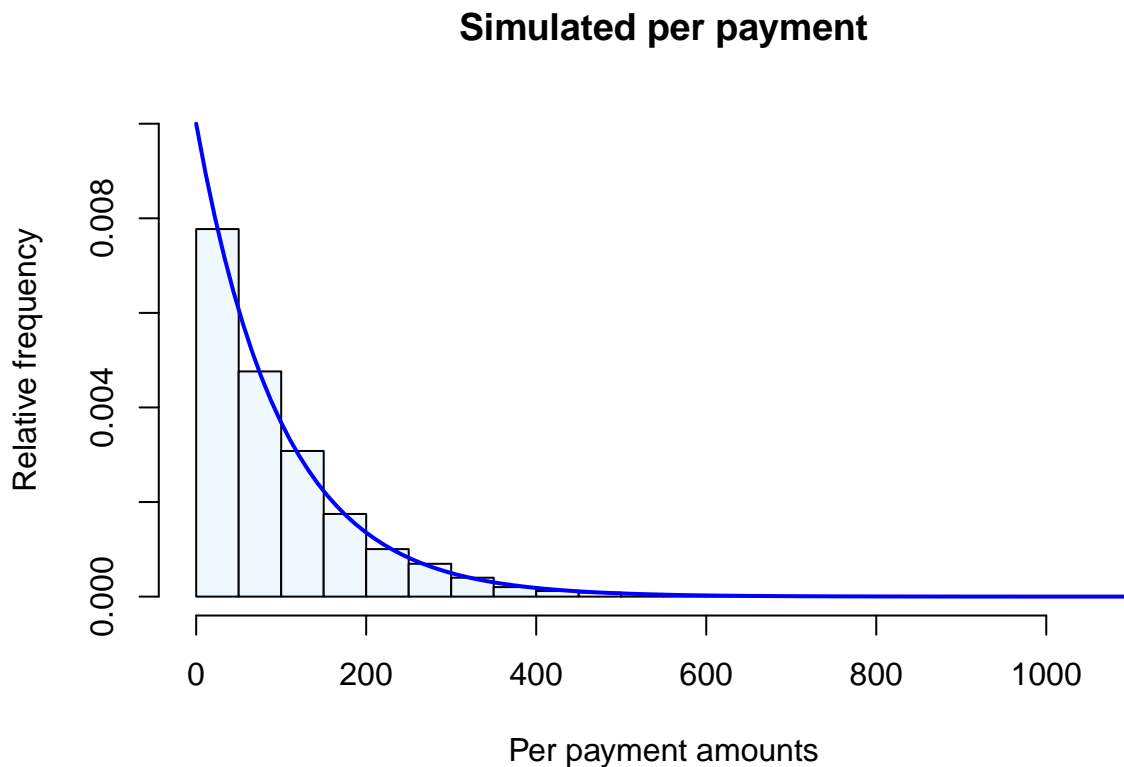
```
mean(y.p)
```

```
## [1] 99.68224
```

What do you notice?

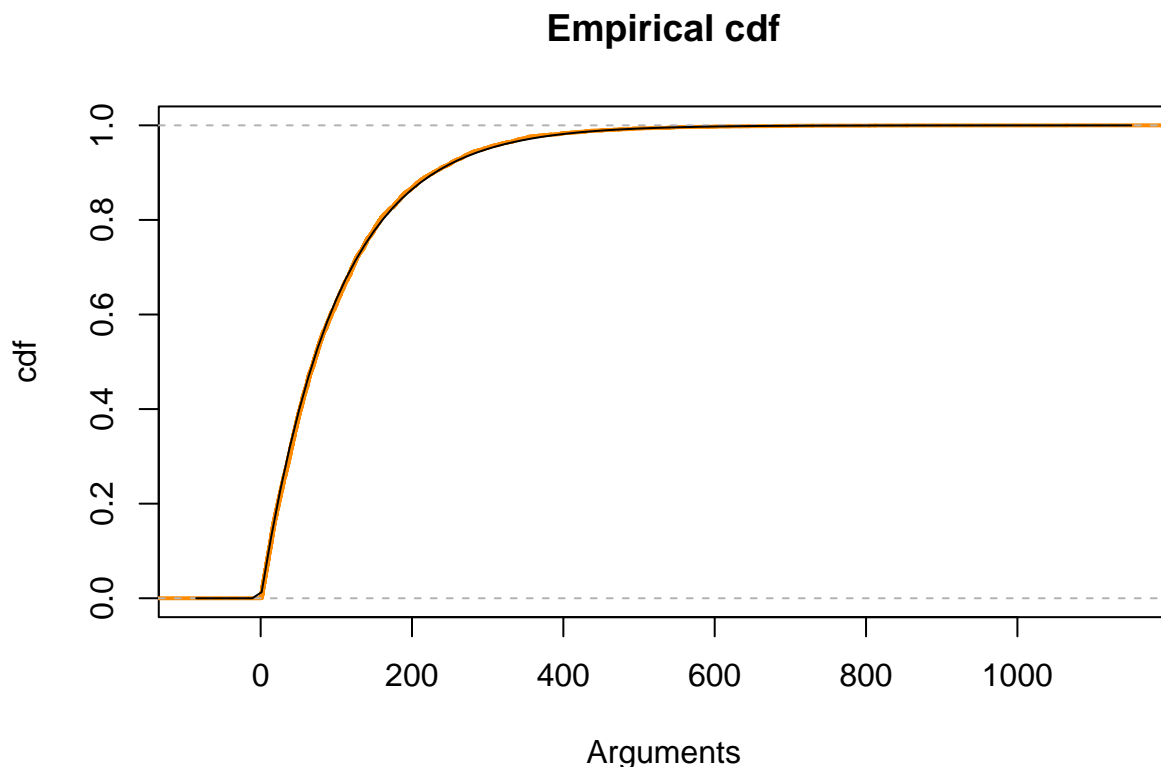
Let's plot the histogram and superimpose the theoretical density. Remember the **memoryless property** of the exponential distribution.

```
hist(y.p, breaks=25, ylim=c(0, dexp(0, rate=1/theta)),  
      main="Simulated per payment",  
      xlab="Per payment amounts",  
      ylab="Relative frequency",  
      prob=TRUE, col="aliceblue")  
curve(dexp(x, rate=1/theta), col="blue", lwd=2, add=TRUE)
```



We can do a similar thing with the empirical cumulative distribution function.

```
cum.y.p=ecdf(y.p)  
plot(cum.y.p,  
      main="Empirical cdf",  
      xlab="Arguments",  
      ylab="cdf",  
      col="darkorange", lwd=2)  
curve(pexp(x, rate=1/theta), add=TRUE)
```



Note

that there is evidence that the random variable Y^P is again continuous.

Per loss

Again luckily, R is versatile enough so that we can create the draws of the per loss random variable pretty easily based on the simulated losses.

```
y.l=(sims-d)*(sims>d)
```

You can make sure that the vector is still `nsim` long.

```
length(y.l)
```

```
## [1] 10000
```

There is another way to construct the per loss random variable in R. Note the use of the command `pmax`.

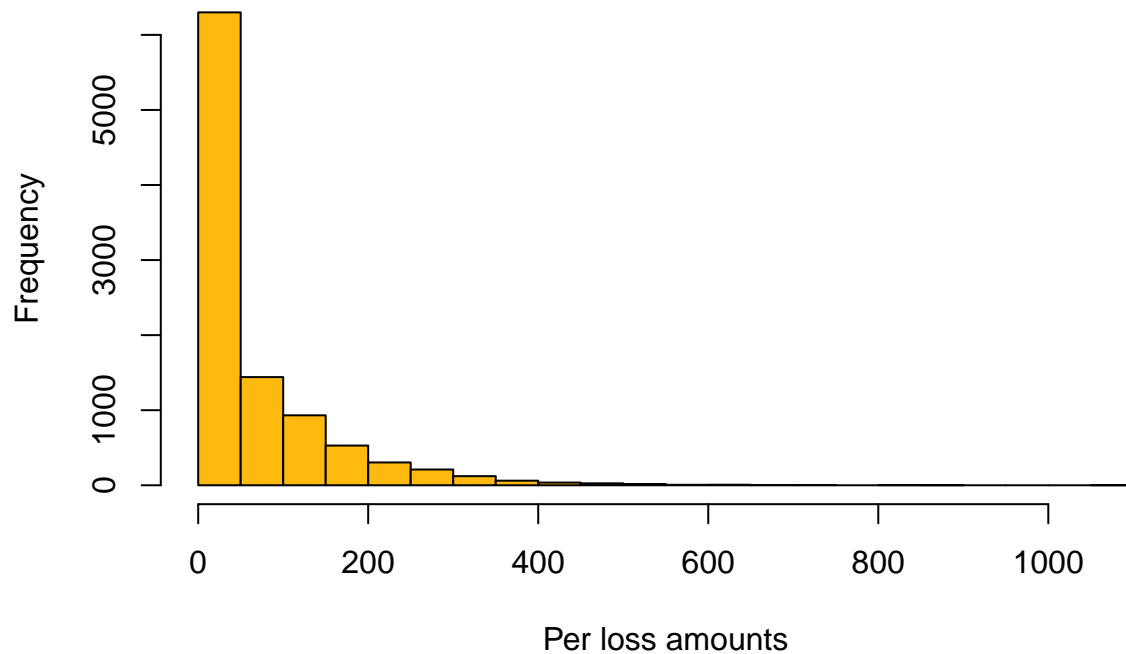
```
y.l.other=pmax(sims-d,0)
sum(y.l==y.l.other)
```

```
## [1] 10000
```

Here is the histogram of the simulated values of the per loss random variable.

```
hist(y.l, breaks=25,
      main="Simulated per loss",
      xlab="Per loss amounts",
      ylab="Frequency",
      col="darkgoldenrod1")
```

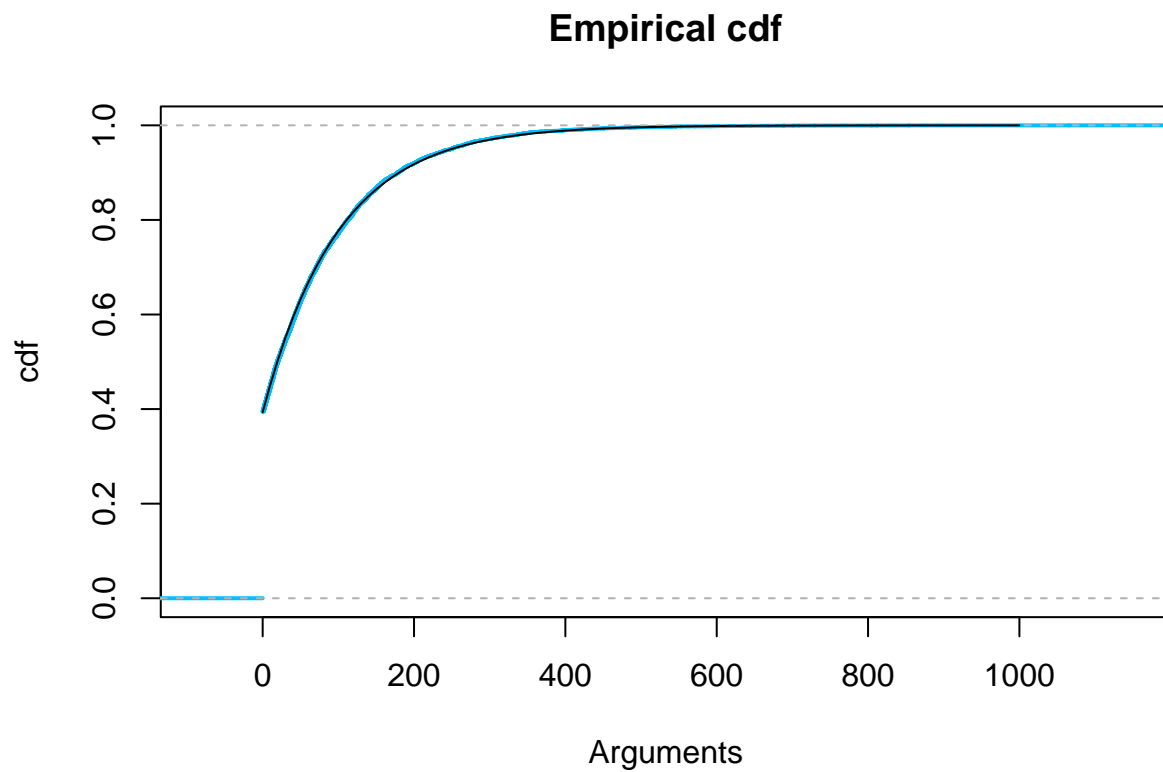
Simulated per loss



Note the tall bar at the far left!

The effect of keeping the zeros for all losses when the deductible was not met is even more evident when you consider the empirical cumulative distribution function.

```
cum.y.l=ecdf(y.l)
plot(cum.y.l,
     main="Empirical cdf",
     xlab="Arguments",
     ylab="cdf",
     col="deepskyblue", lwd=2)
curve(pexp(x+d, rate=1/theta), xlim=c(0,1000), add=TRUE)
```



Note the jump at zero!

Limited loss

Finally, we consider the limited loss random variable.

```
x.d=sims*(sims<=d)+d*(sims>d)
length(x.d)
```

```
## [1] 10000
```

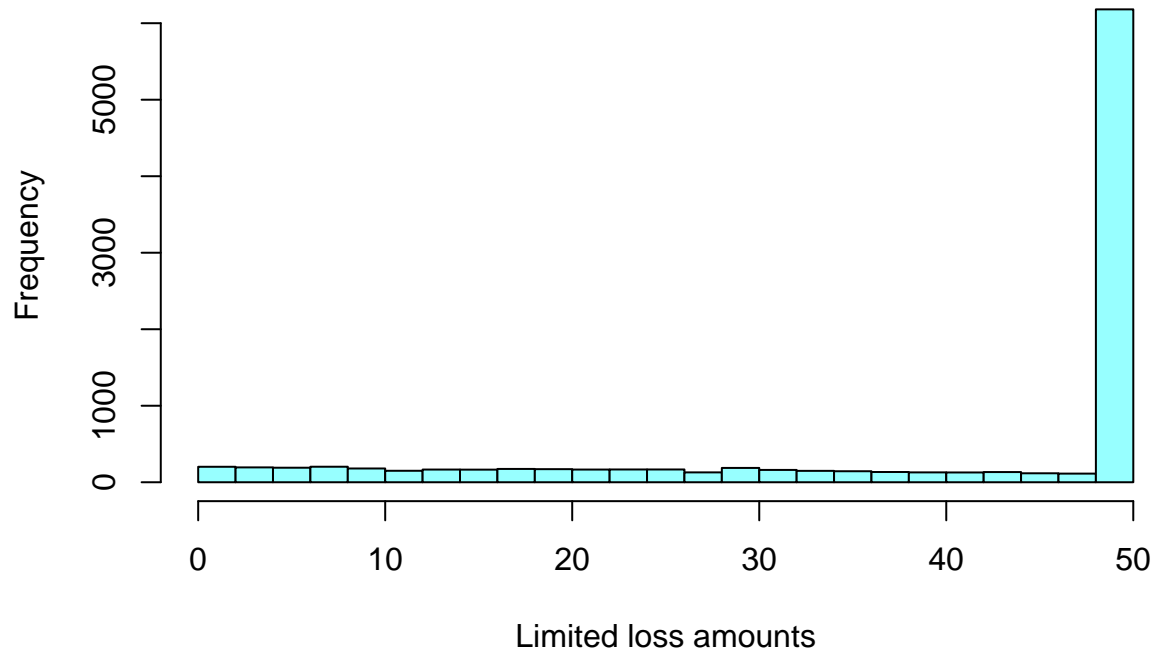
```
x.d.other=pmin(sims,d)
sum(x.d==x.d.other)
```

```
## [1] 10000
```

Here is the histogram of the limited loss random variable.

```
hist(x.d, breaks=25,
      main="Simulated limited loss",
      xlab="Limited loss amounts",
      ylab="Frequency",
      col="darkslategray1")
```

Simulated limited loss



Note the tower at 50!

Again, it is useful to look at the empirical cumulative distribution function.

```
cum.x.d=ecdf(x.d)
plot(cum.x.d, xlim=c(0, 900),
     main="Empirical cdf",
     xlab="Arguments",
     ylab="cdf",
     col="darkorchid", lwd=2)
curve(pexp(x, rate=1/theta), xlim=c(0,d), add=TRUE)
```


Empirical cdf

