Tutorial: Poisson and compound Poisson processes

Almut Veraart

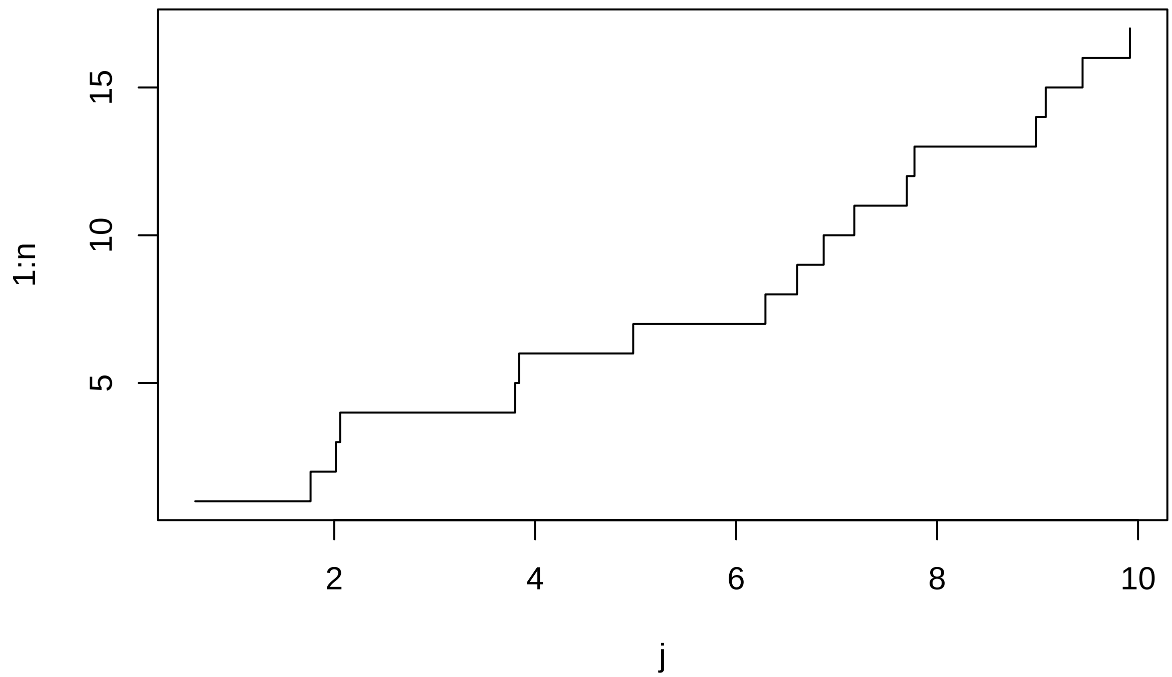
Autumn 2022

Poisson processes

We can simulate a Poisson process with rate $\lambda > 0$ on the time interval $[0, t]$, using the results from the lectures, as follows:

1) We first want to find the number of jumps at time t, so we simulate a random variable with $\text{Poi}(\lambda t)$ distribution.
2) Next we want to know when these jumps occur. So we simulate the jump times from the uniform distribution on $[0, t]$ and then order these jump times (so that the first jump occurs first etc.).

```
set.seed(1)
t <- 10 # generate the process on the time interval [0,t]
lambda <- 2 # rate of the process
n <- rpois(1,lambda*t) #Find the number of jumps on [0,t]
j <- sort(runif(n,0,t)) #Simulate uniform jump times and order them.

plot(j,1:n,type="s")  # Make a staircase plot
```

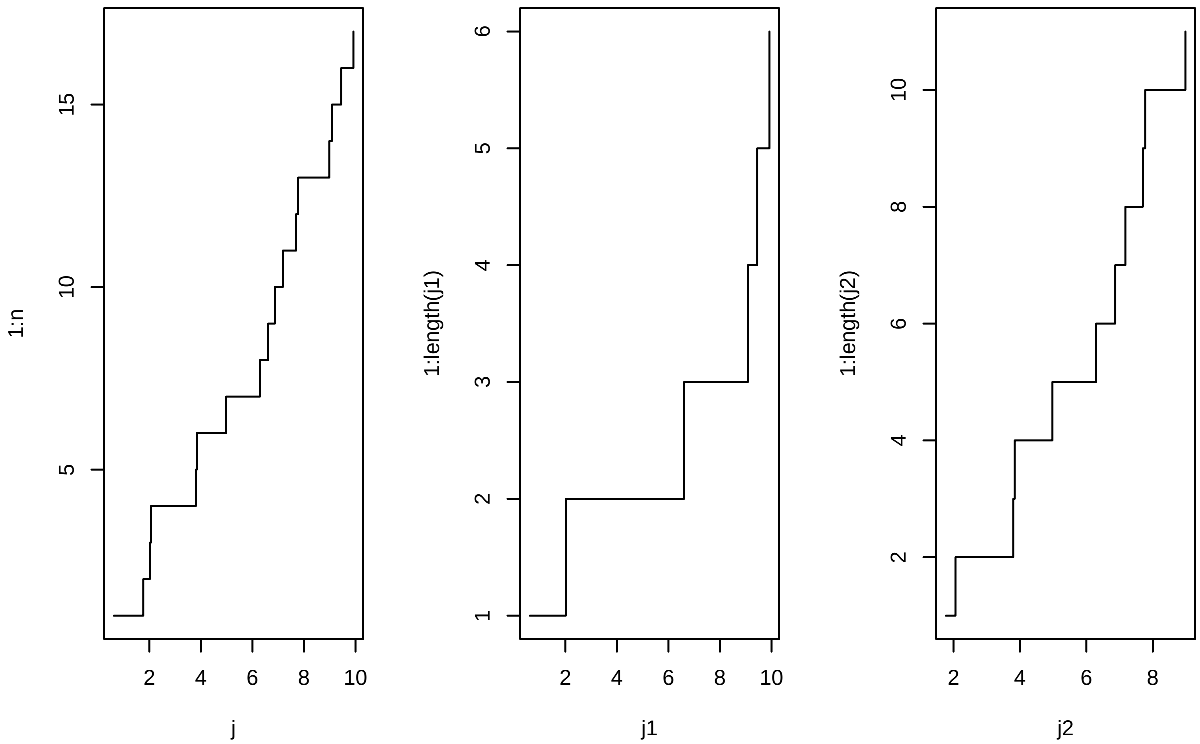


Thinning of a Poisson process

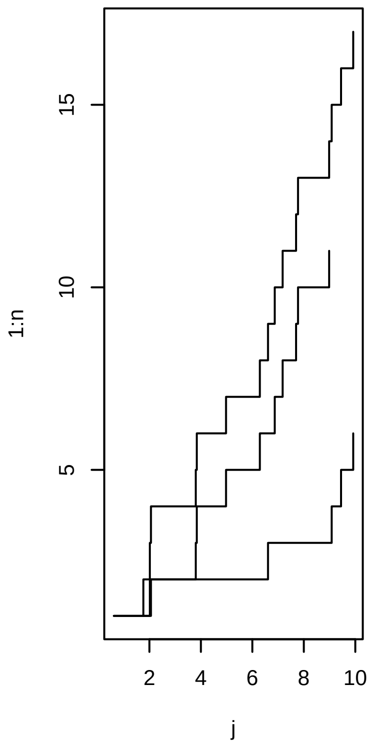
Let us now consider the thinning of a Poisson process. We simulate the Poisson process as above and split it into two parts according to the probabilities p and $1-p$.

We can then plot the original Poisson process and the two thinned Poisson processes.

```
# Thinning of a Poisson process
p <- 0.4
y <- rbinom(n,1,p)
j1 <- j[y==1]
j2 <- j[y==0]

par(mfrow=c(1,3))
plot(j,1:n,type="s")
plot(j1,1:length(j1),type="s")
plot(j2,1:length(j2),type="s")
```




```
plot(j,1:n,type="s")
lines(j1,1:length(j1),type="s")
lines(j2,1:length(j2),type="s")
```



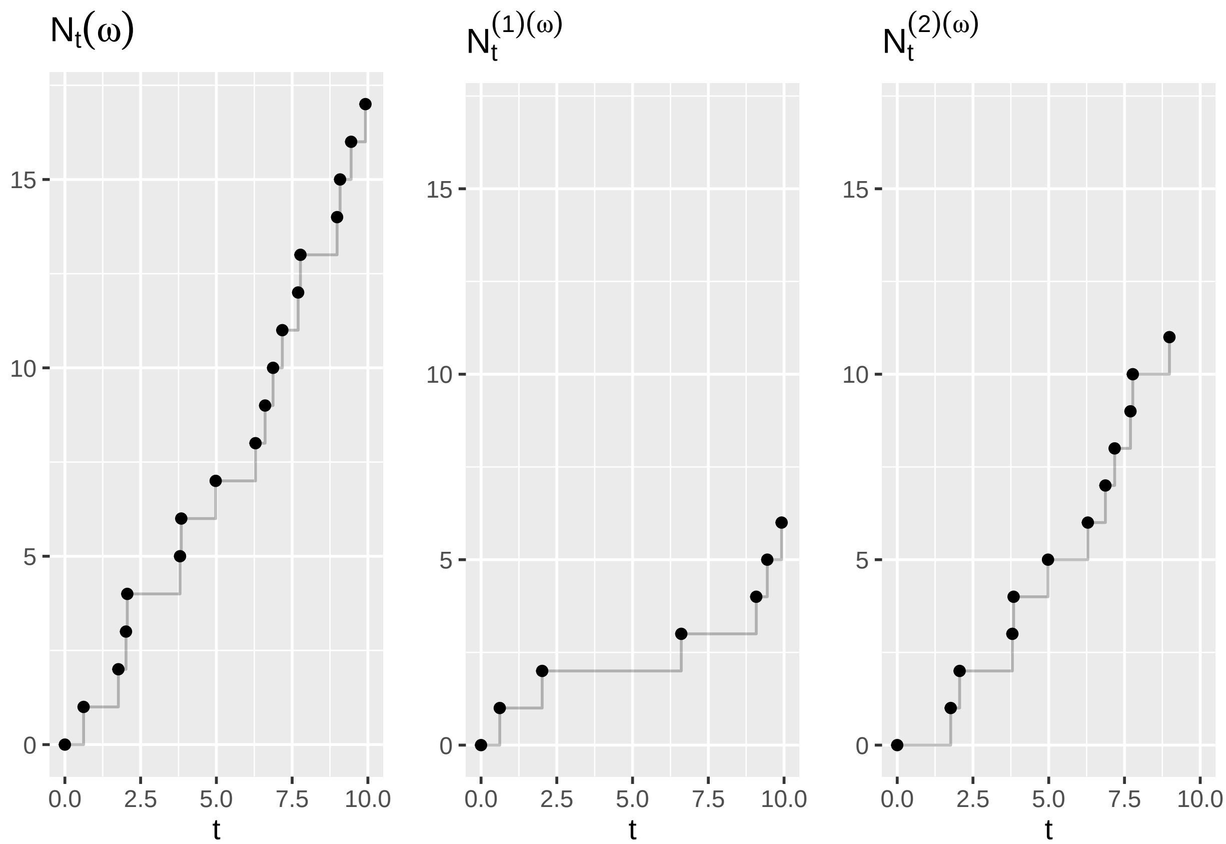
Reproducing the plots from the lecture notes

If you are interested in re-producing the plots from the lecture notes, you can work through the following code (using the ggplot2 package again).

```
library(ggplot2)  #For very pretty plots
library(latex2exp) #For LaTex annotations in the graphs
library(gridExtra) #For combining several plots in one picture
library(scales)
```

```
alljumps <- c(0,seq(1, n))
p1jumps <- c(0,seq(1,length(j1)))
p2jumps <- c(0,seq(1,length(j2)))


q<-qplot(c(0,j), alljumps) +
  geom_step(alpha = 0.25)+
labs( title = TeX("$N_t(\\omega)$ "),x=TeX("$t$"),y=TeX(" ")) +
    coord_cartesian(xlim = c(0, t), ylim=c(0,n))

q1<-qplot(c(0,j1), p1jumps) +
  geom_step(alpha = 0.25)+
  labs( title = TeX("$N^{(1)}_{t}(\\omega)$ "),x=TeX("$t$"),y=TeX(" ")) +
  coord_cartesian(xlim = c(0, t), ylim=c(0,n))

q2<-qplot(c(0,j2), p2jumps) +
```

```
  geom_step(alpha = 0.25)+
  labs( title = TeX("$N^{(2)}_{t}(\\omega)$ "),x=TeX("$t$"),y=TeX(" ")) +
  coord_cartesian(xlim = c(0, t), ylim=c(0,n))

grid.arrange(q, q1, q2, ncol=3)
```



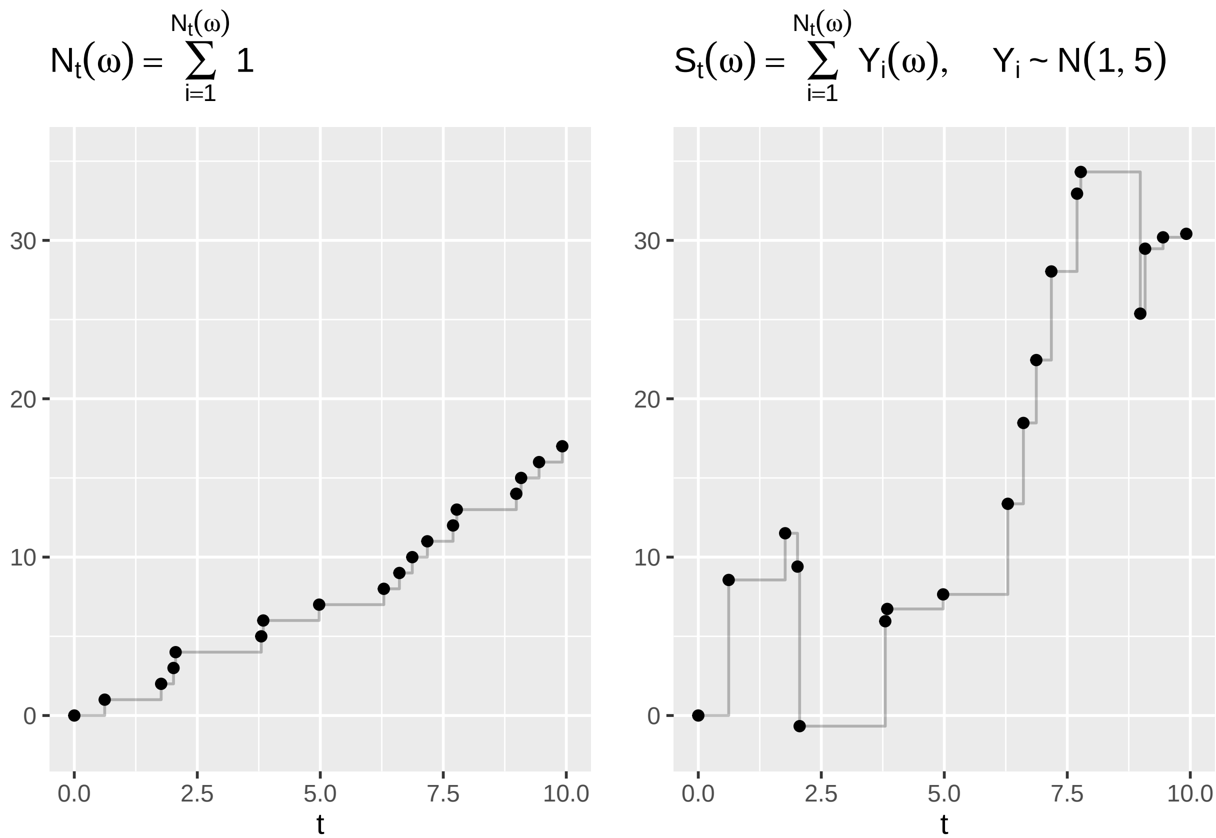
Compound Poisson process

Let us now simulate a compound Poisson process.

First we simulate the number of jumps

```
###Simulate compound Poisson process
set.seed(1)
t <- 10
lambda <- 2
N <- rpois(1, lambda * t)
events <- sort(runif(N,0,t))

#First, we consider a vector of jumps of size 1 (to simulate a Poisson process)
jumps<-seq(1, N)
#njumps<-numeric(N)

#Second, we simulated normally distributed jump sizes
jumpsizes <-rnorm(N,1,5)
njumps<-cumsum(jumpsizes)
```

```r
ymax <-max(c(N,njumps))+1
ymin <-min(c(0,njumps))-1

#We plot the Poisson process
cp1 <- qplot(c(0,events), c(0,jumps)) +
  geom_step(alpha = 0.25)+
  labs( title = TeX("$N_{t}(\\omega)=\\sum_{i=1}^{N_{t}(\\omega)} 1$ "),x=TeX("$t$"),y=TeX(" ")) +
  coord_cartesian(xlim = c(0, t), ylim=c(ymin,ymax))

#and the compound Poisson process
cp2 <- qplot(c(0,events), c(0,njumps)) +
  geom_step(alpha = 0.25)+
  labs( title = TeX("$S_t(\\omega)=\\sum_{i=1}^{N_{t}(\\omega)}Y_i(\\omega), \\; \\; Y_i  \\sim N(1,5)$
  coord_cartesian(xlim = c(0, t), ylim=c(ymin,ymax))

grid.arrange(cp1, cp2, ncol=2)
```



The Cramer Lundberg model

Let us now consider the Cramer Lundberg model we discussed in the lectures.

In that model, we consider a Poisson process $N = (N_t)_{t\geq 0}$ of rate $\lambda > 0$ and a sequence of i.i.d.~positive random variables $Y = (Y_j)_{j\in\mathbb{N}}$. We assume that N and Y are independent. Then we define the compound

Poisson process $(S_t)_{t\geq 0}$ as

$$S_t = \sum_{j=1}^{N_t} Y_j,$$

which describes the **total claim amount**.

Further, the **risk process** $\{U_t\}_{t\geq 0}$ is defined as

$$U_t = u + ct - S_t, \qquad t \geq 0,$$

where $u \geq 0$ stands for the **initial capital** and $c > 0$ denotes the **premium income rate**.

We will now show how we can simulate the risk process.

First we write a function for simulating a compound Poisson process on the interval $[0, T]$. Here we simulate the process on a discrete grid with grid length δ.

```
#Interval [0,T], grid length delta
#lambda = parameter of Poisson process
#rate = parameter of exponential distribution
CompoundPoisson_Exp_Sim <-function(T=10, delta=0.0001, lambda =1, rate=1 )
{
  pois_incr <- rpois(T/delta,lambda*delta)

  s_incr <- numeric(T/delta)
  for(i in 1:(T/delta)){

    if(pois_incr[i]>0){
      s_incr[i]<-sum(rexp(pois_incr[i], rate=rate))
    }

  }

  s <- numeric(T/delta+1)
  s[2:(T/delta+1)]<-cumsum(s_incr)

  s
}
```

Next we write a function to simulate the drift part of the Cramer Lundberg risk process.

```
#Interval [0,T], grid length delta
#c premium, u initial wealth
CL_Drift_Sim <-function(T=10, delta=0.0001, c=10, u=10 )
{

  drift_incr <- rep(c*delta,T/delta)

  drift <- numeric(T/delta+1)
  drift[1]<-u
  drift[2:(T/delta+1)]<- u+cumsum(drift_incr)
  drift
}
```

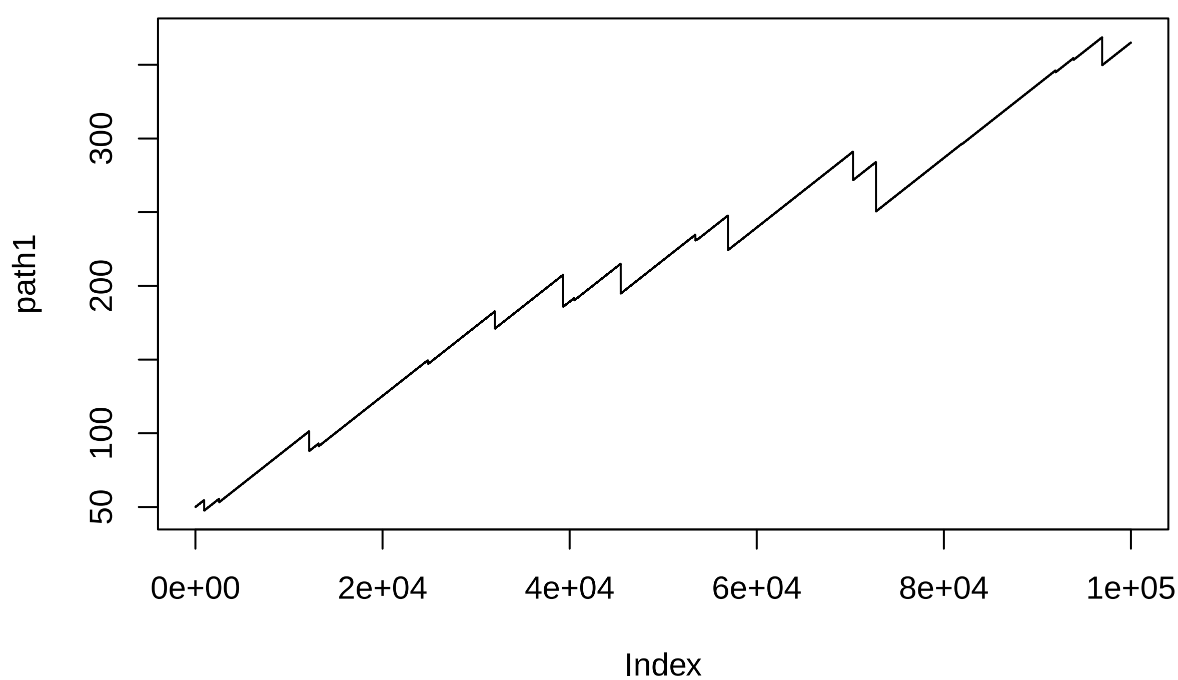
We can now simulate the risk process as follows.

```
#Simulate the risk process U
#Interval [0,T], grid length delta
```

```
#lambda = parameter of Poisson process
#rate = parameter of exponential distribution
#c premium, u initial wealth
CL_U_Sim <- function(T=10, delta=0.0001, lambda =1, rate=1,c=10, u=10){
  drift <- CL_Drift_Sim(T,delta,c,u)
  s <- CompoundPoisson_Exp_Sim(T, delta, lambda, rate)
  U<-drift-s
  trunc_U <- U

  if(length(which(U<0))>0){
    indx <- min(which(U<0))
    if(indx>0){
      for(i in indx:(length(U))){
        trunc_U[i]<-0
      }
    }
  }
  trunc_U

}
```

We can now simulate a path of the risk process and plot it:

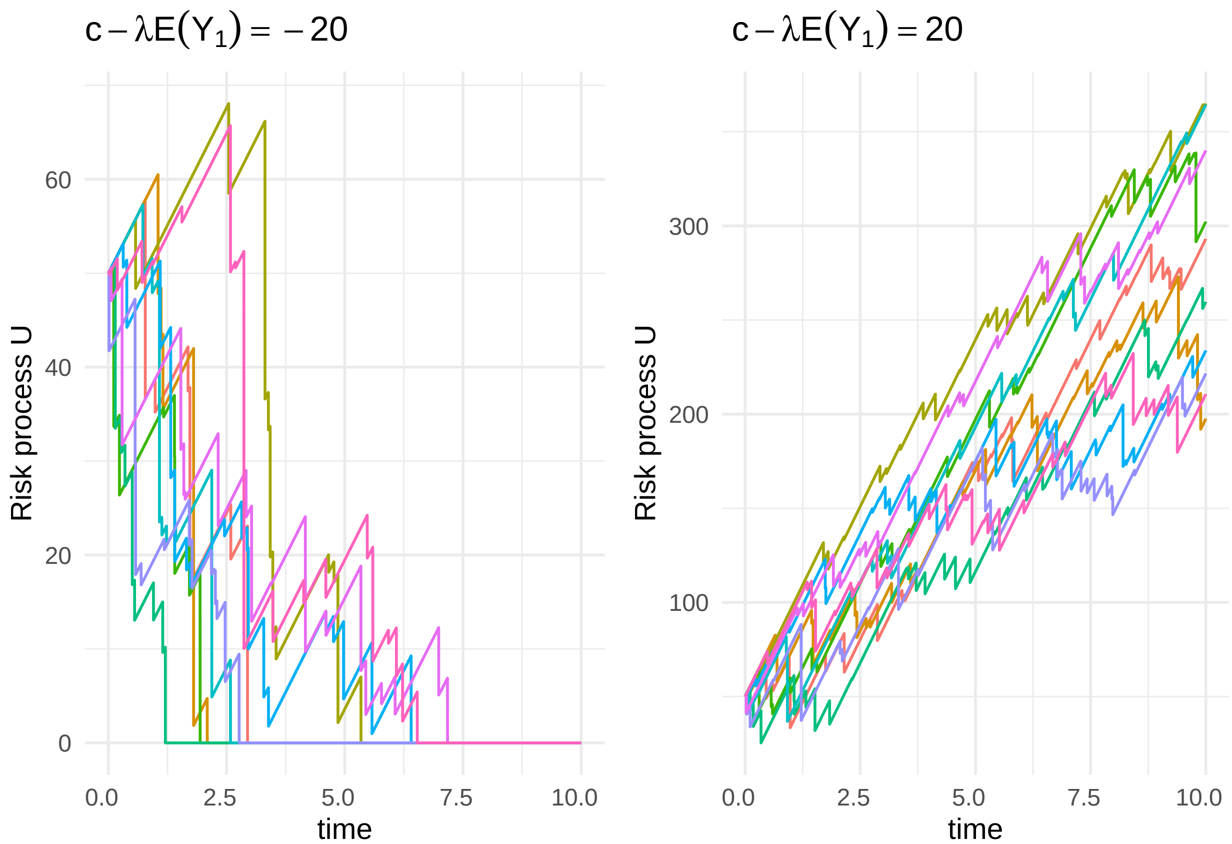
```
path1<-CL_U_Sim(T=10, delta=0.0001, lambda =3, rate=0.1,c=50, u=50)
plot(path1, type="l")
```



Reproducing the pictures from the lecture notes

You can reproduce the pictures from the lecture notes as follows.

```
#Create a data set of sample paths
set.seed(1)
T<-10
delta <-0.001
NPaths <-10
length <-T/delta+1

data1 <-matrix(0,nrow=NPaths, ncol=length)
data2 <-matrix(0,nrow=NPaths, ncol=length)



for(i in 1:NPaths)
{
  data1[i,]<-CL_U_Sim(T=10, delta=delta, lambda =3, rate=0.1,c=10, u=50)
}

theme_set(theme_minimal())
p1<-qplot(c(col(data1)), c(data1), group = c(row(data1)), colour = factor(c(row(data1))),
          geom = "line",main= TeX("$c-\\lambda E(Y_1)=-20$"), xlab="time", ylab="Risk process U",show.le
p1b<-p1+scale_x_continuous(labels = unit_format(unit = " ", scale = delta))

######


for(i in 1:NPaths)
{
  data2[i,]<-CL_U_Sim(T=10, delta=delta, lambda =3, rate=0.1,c=50, u=50)
}


theme_set(theme_minimal())
p2<-qplot(c(col(data2)), c(data2), group = c(row(data2)), colour = factor(c(row(data2))),
          geom = "line",main= TeX(" $c-\\lambda E(Y_1)=20$"), xlab="time", ylab="Risk process U",show.le
p2b<-p2+scale_x_continuous(labels = unit_format(unit = " ", scale = delta))

grid.arrange(p1b, p2b, ncol=2)
```



$c - \lambda E(Y_1) = -20$
Risk process U
0
20
40
60
0.0
2.5
5.0
7.5
10.0
time
$c - \lambda E(Y_1) = 20$
Risk process U
100
200
300
0.0
2.5
5.0
7.5
10.0
time