Lab8

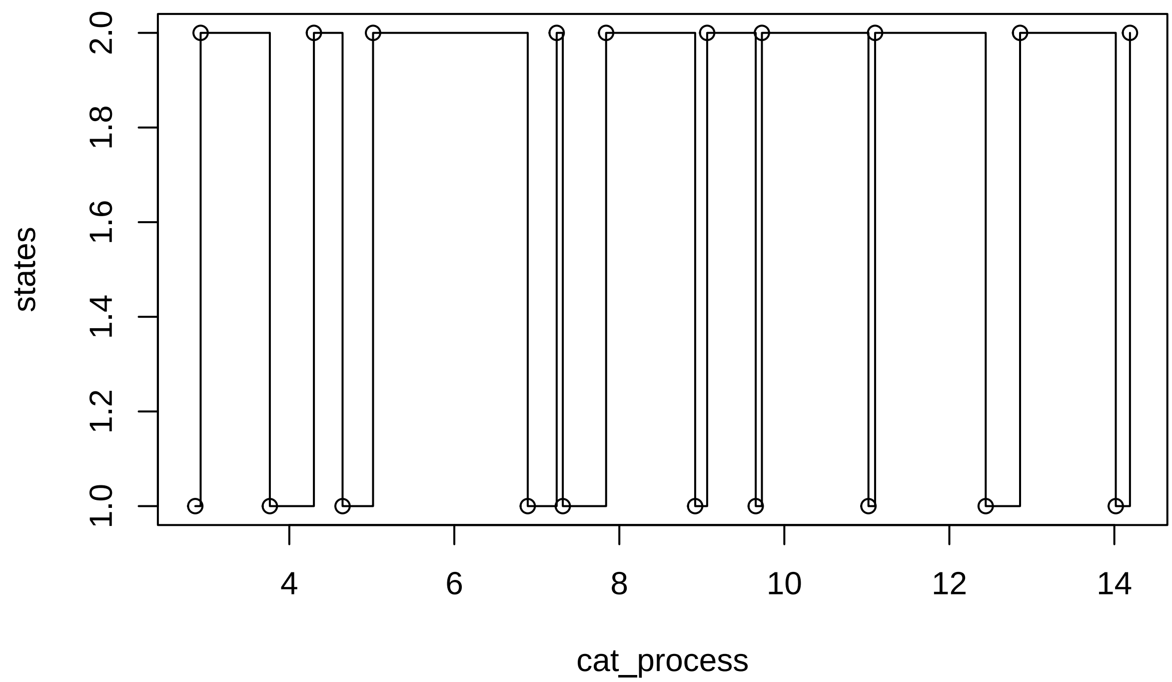
Generating Markov process with continuous state space and Monte Carlo methods

Piotr Ginalski

2025-04-14

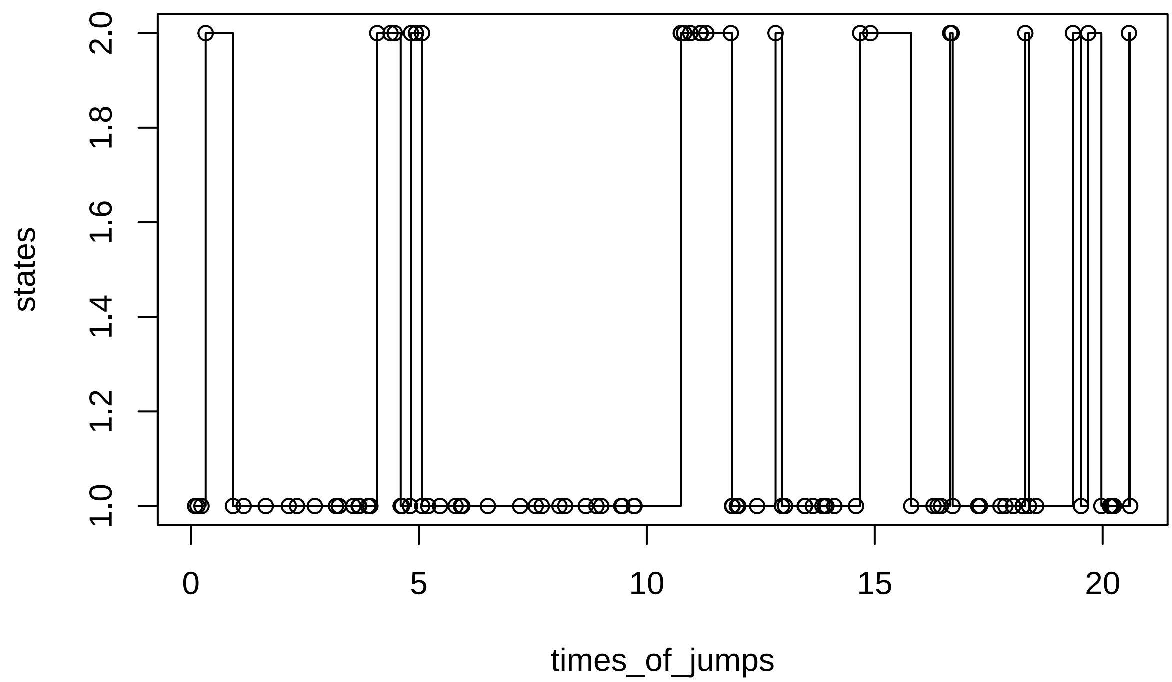
Let's generate Markov process with continuous state space. We are going to simulate two Poisson processes - one for Poisson(alpha) and second for Poisson(beta). From them we are going to simulate final process and plot the trajectory.

```
alpha <- 1
beta <- 3
n_jumps <- 20
n_jumps_one <- n_jumps / 2

first_process <- rexp(n_jumps_one, rate = alpha)
second_process <- rexp(n_jumps_one, rate = beta)
cat_process <- numeric(n_jumps)
cat_process[1] <- first_process[1]
cat_process[2] <- second_process[1] + first_process[1]

for(i in (2:n_jumps_one)){
  cat_process[2*i - 1] = cat_process[2*i - 2] +first_process[i]
  cat_process[2*i] = cat_process[2*i - 1] + second_process[i]
}

states <- rep(c(1,2), n_jumps_one)

{
plot(cat_process, states, type = "s")
points(cat_process, states)
}
```



And now we are going to generate Markov process by using "virtual jumps".

```
rate <- 5
alpha <- 1
beta <- 3

n_virtual_jumps <- 100

times_of_jumps <- cumsum(rexp(n_virtual_jumps, rate))
states <- numeric(n_virtual_jumps)
states[1] <- 1

for(i in (2:n_virtual_jumps)){
  if (states[i-1] == 1){
    states[i] <- ifelse(runif(1) < alpha/rate, 2, 1)
  }

  if (states[i-1] == 2){
    states[i] <- ifelse(runif(1) < beta/rate, 1, 2)
  }
}

{
plot(times_of_jumps, states, type = "s")
points(times_of_jumps, states)
}
```



Let's clear env.

```
rm(list = ls())
```

And now we are going to use Monte Carlo method to calculate integral of Cauchy density (from 2 to inf).

```
1 - pcauchy(2)
```

```
## [1] 0.1475836
```

```
cauchy_simulation <- function(repeats){

  estimate <- mean(rcauchy(repeats) > 2)
  sig <- sd(rcauchy(repeats) > 2) / sqrt(repeats)

  return(c(estimate, sig))
}

n_sim <- 200
simulated_value <- numeric(n_sim)
for(i in (1:n_sim)){
  simulated_value[i] <- cauchy_simulation(100)[1]
}

cauchy_simulation(100)
```

```
## [1] 0.1500000 0.0314466
```

We can also do MC simulation by using other estimations of mean. We are doing that to reduce variance.

```
cauchy_simulation_2 <- function(repeats){

  estimate <- (1/2) * mean(abs(rcauchy(repeats)) > 2)
  sig <- sd(abs(rcauchy(repeats)) > 2) / (2*sqrt(repeats))

  return(c(estimate, sig))
}

cauchy_simulation_2(100)
```

```
## [1] 0.19000000 0.02324116
```