Lab 6

Markov Chains

Piotr Ginalski

In this laboratory, we are going to consider simulating Markov Chain. Let's consider easy example of Random Walk with absorbing barriers.

```r
simulate_absorbing_walk <- function(x0, k, p){
  x <- x0
  trajectory <- x0
  while (x > 0 &&  x < k){
    if (runif(1) < p){
      x <- x + 1
      trajectory <- c(trajectory, x)
    }
    else{
      x <- x - 1
      trajectory <- c(trajectory, x)
    }
  }
  return(trajectory)
}


simulate_absorption <- function(x0, k, p, n_sim) {
  count0 <- 0
  countK <- 0

  for (i in 1:n_sim) {
    trajectory <- simulate_absorbing_walk(x0, k, p)
    final_state <- trajectory[length(trajectory)]
    if (final_state == 0) {
      count0 <- count0 + 1
    } else {
      countK <- countK + 1
    }
  }

  psi0 <- count0 / n_sim
  psiK <- countK / n_sim
  return(c(psi0 = psi0, psiK = psiK))
}

simulate_absorption(5, 20, 0.4, 10^5)
```

```
##    psi0    psiK
```

```
## 0.99824 0.00176
```

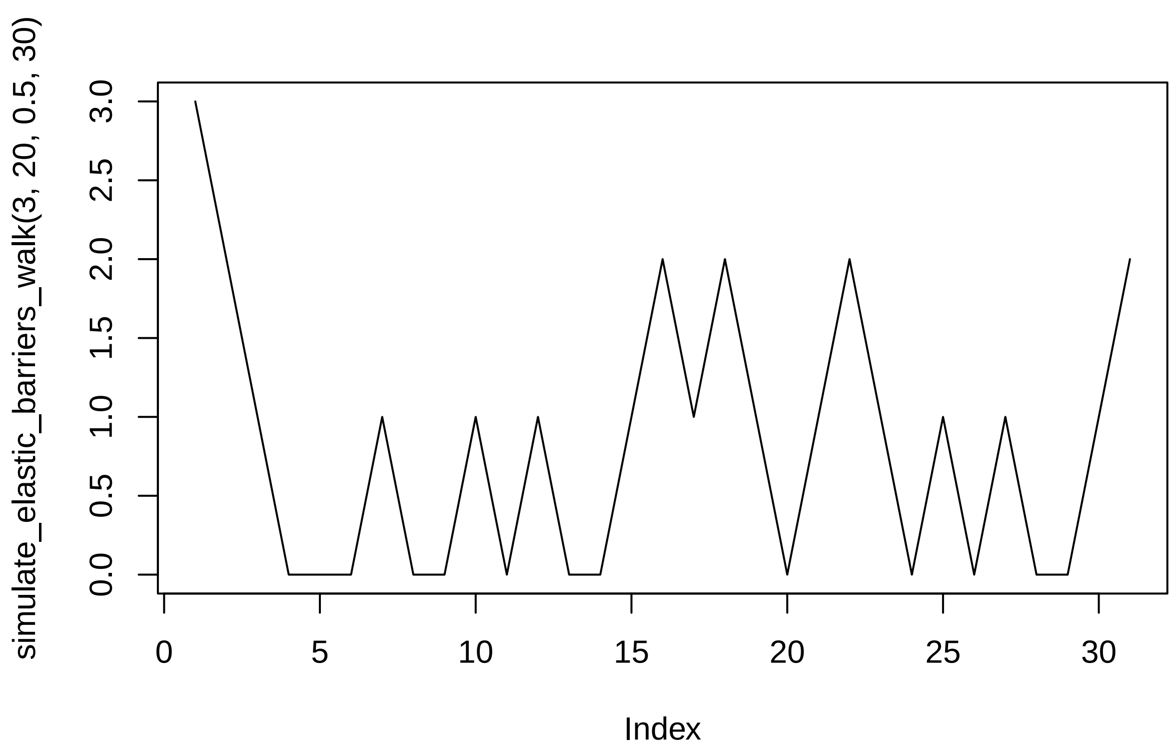
```r
simulate_absorption(15, 20, 0.4, 10^5)
```

```
##    psi0    psiK
## 0.86766 0.13234
```

And now we create random walk with elastic barriers.

```r
simulate_elastic_barriers_walk <- function(x0, k, p, n){
  x <- x0
  trajectory <- x0
  for (i in 1:n){
    if (runif(1) < p){
      x <- min(x+1, k)
      trajectory <- c(trajectory, x)
    }
    else{
      x <- max(x-1, 0)
      trajectory <- c(trajectory, x)
    }
  }
  return(trajectory)
}

plot(simulate_elastic_barriers_walk(3, 20, 0.5, 30), type = "l")
```



```r
simulate_asymptotic <- function(x0, k, p, n, x){
  trajectory <- simulate_elastic_barriers_walk(x0, k, p, n)
  visits <- sum(trajectory == x)
  return (visits / n)
}


p <- 0.3
q <- 1 - p
x <- 0
k <- 20


# We can compare our simulation with theoretical formula
print(simulate_asymptotic(x0 = 5, k = 20, p, 10000, x))
```

```
## [1] 0.5869
```

```r
print(( (1 - p/q) / (1 - (p/q)^(k+1))) * (p/q)^(x) )
```

```
## [1] 0.5714286
```

Let's do next exercise. We are going to simulate Markov Chain with continuous state space.

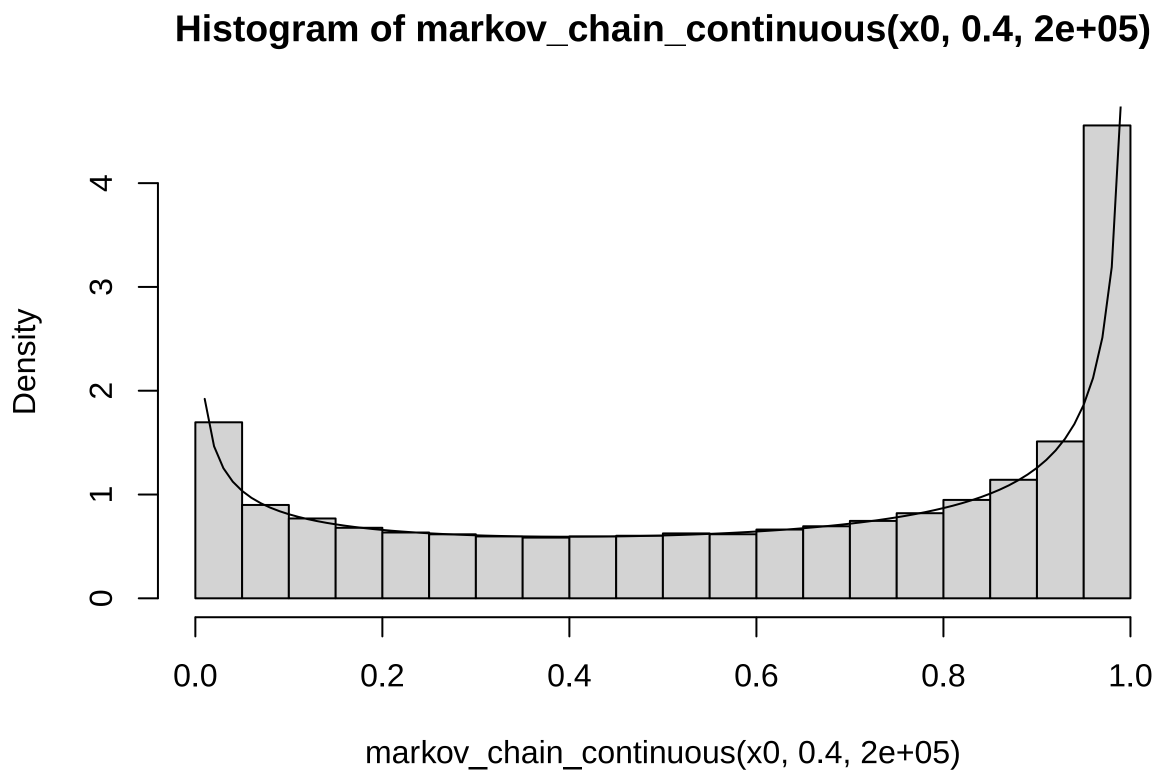
```r
a <- 0.4
n_time <- 20
x0 <- 0.5

markov_chain_continuous <- function(x0, a, n_time){
  trajectory <- numeric(n_time + 1)
  trajectory[1] <- x0
  for (i in 2:(n_time + 1)){
  if (runif(1) < a){
    x <- runif(1, min = 0, max = trajectory[i - 1])
    trajectory[i] <- x
  }
  else{
    x <- runif(1, min = trajectory[i - 1], max = 1)
    trajectory[i] <- x
  }
  prev_x <- x
}
return(trajectory)
}

markov_chain_continuous(x0, 0.4, 20)
```

```
##  [1] 0.5000000000 0.6133402643 0.0002411445 0.6536929508 0.6406453595
##  [6] 0.8250584420 0.9520956243 0.2499687693 0.8119224457 0.7852863522
## [11] 0.6317969265 0.8056438451 0.8405976847 0.9801336942 0.9985893049
## [16] 0.6982901419 0.0868458508 0.0093749333 0.0074237760 0.0175732139
## [21] 0.7997305389
```

```r
x0 <- 0.1
{
hist(markov_chain_continuous(x0, 0.4, 200000), probability = TRUE)
curve(dbeta(x, shape1 = 0.6, shape2 = 0.4), add= TRUE)
}
```



As a homework, we can do problem 2.10 from lecture notes.

```r
n_time <- 20
x0 <- 1/2
a <- 1/3
```