

Project 2017

Lucien Ledune

1 dÃ©cembre 2017

Introduction

In this project, we are going to discuss the estimation of the cumulative distribution and the density function of an exponentially distributed random variable. We will consider two different cases for the density estimator, 2nd order and 4th order kernels.

Then we will perform a Monte Carlo simulation on the IMSE of our estimators. Our goal here will be to investigate the rates of convergence of the three estimators.

Preparation of data

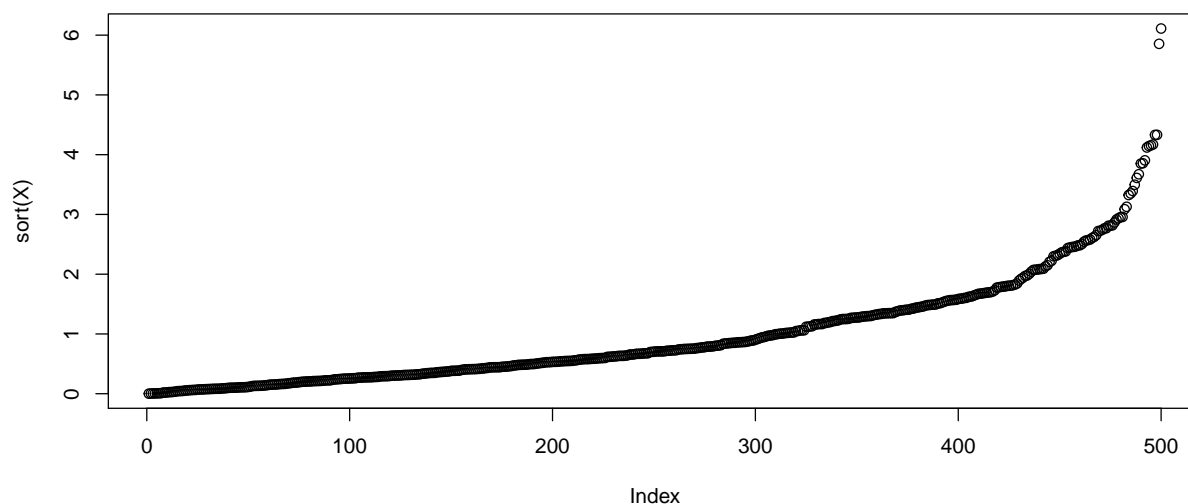
In order to simulate a random variable, we will generate X points using `rexp()`. This will give us points following the exponential distribution.

We can plot the (ordered) estimated data here and make sure it follows an exponential distribution.

```
set.seed(856)
```

```
X = rexp(500)
```

```
plot(sort(X))
```



Nonparametric estimation of the cumulative density function

If we want to determine what the function's cumulative density function is, a simple way to do that is to use the given points to calculate the following distribution :

- 0 if $x \leq X_{(1)}$
- k/n if $X_{(k)} \leq x < X_{(k+1)}$ for $k = 1, \dots, n - 1$
- 1 if $x \geq X_{(n)}$

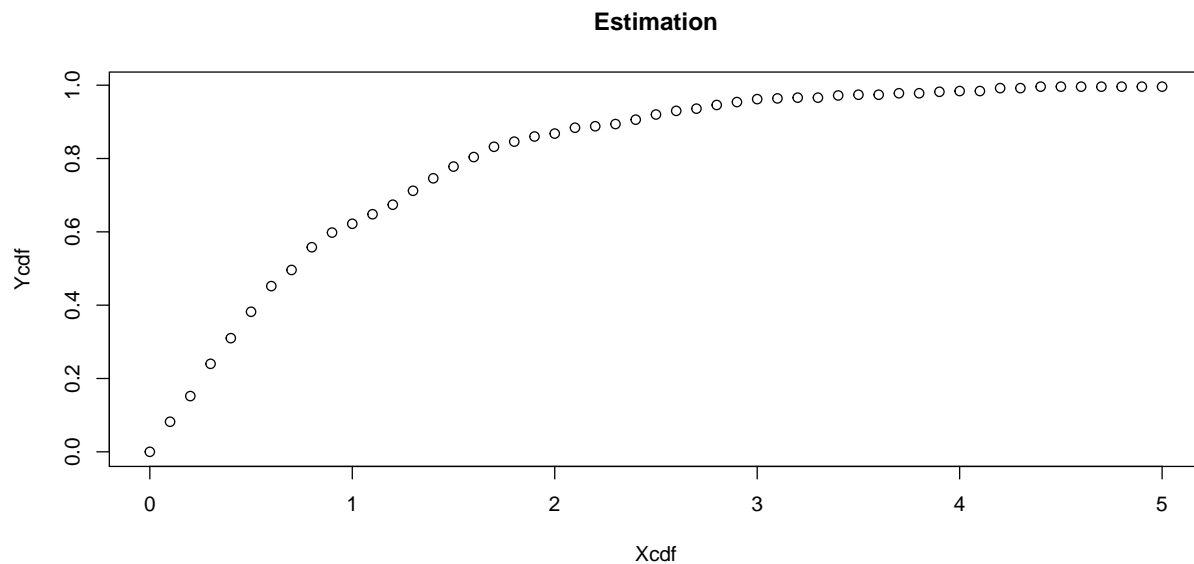
We are basically calculating the proportion of data under a point x .

Let's apply this to our function :

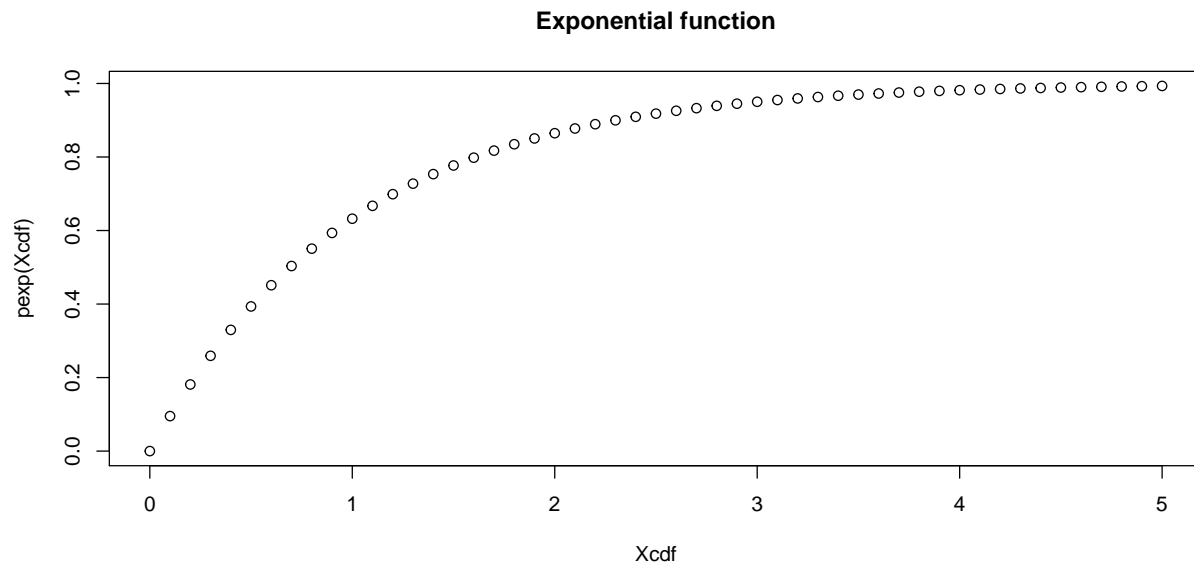
```
Ycdf = NULL
n = length(X)
Xcdf = seq(0,5,0.1)

for(i in 1:length(Xcdf)){
  Ycdf[i] = sum(X < Xcdf[i])/n
}

plot(Xcdf, Ycdf, main = "Estimation")
```



```
plot(Xcdf, pexp(Xcdf), main = "Exponential function")
```



We can easily observe that our ‘non-parametric’ estimation of the exponential cdf is close to reality. This method is very simple and intuitive, but as we can see here it is very good for estimating the cdf of an unknown function.

Nonparametric estimation of density function

We are now going to discuss the density estimation of a function. The simple intuition behind density estimation is that we want to know how many % of the data falls between the interval $[k, k+h]$, h being the wideness of our interval.

To get smoother result, we can use a kernel estimator, it is going to estimate the value of our point at the x points around it. We are basically giving the data points a “continuous” value, which results in a smoother estimation.

$$K = 1/nh \sum_{i=1}^n (x - X_i/h)$$

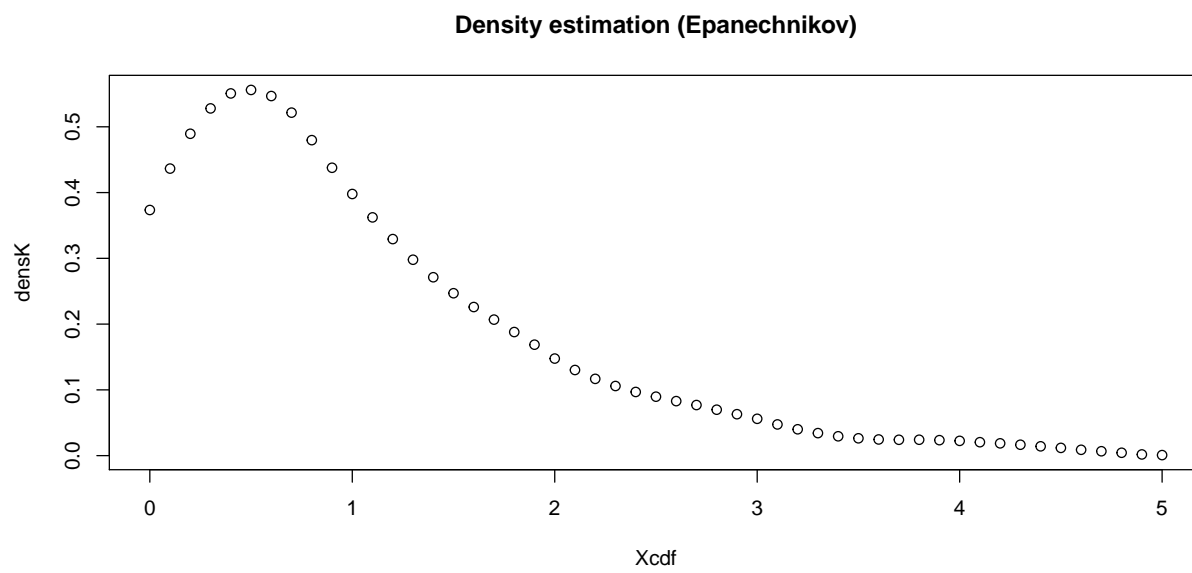
Let’s apply this to our data. First we build the Kernel function and the density estimation function.

```
K = function(u){ 0.75 * (1 - u^2) * (abs(u) <= 1) } #Epanechnikov kernel
DensEst = function(x, X, h, K) mean(K((x - X)/h))/h #Density calculation
h=0.75
```

rajouter ici le choix optimal de h Then we can choose our bandwidth h .

Now we apply the density estimator function to some x points so we can plot it.

```
densK = sapply(Xcdf, function(Xcdf) DensEst(Xcdf, X, h, K))
plot(Xcdf, densK, main = "Density estimation (Epanechnikov)")
```



And we have a function quite similar to the real exponential value (obviously).