Estimation de densité et fonctions de survie dans le cas du modèle de "multiplicative censoring"

Homer Durand

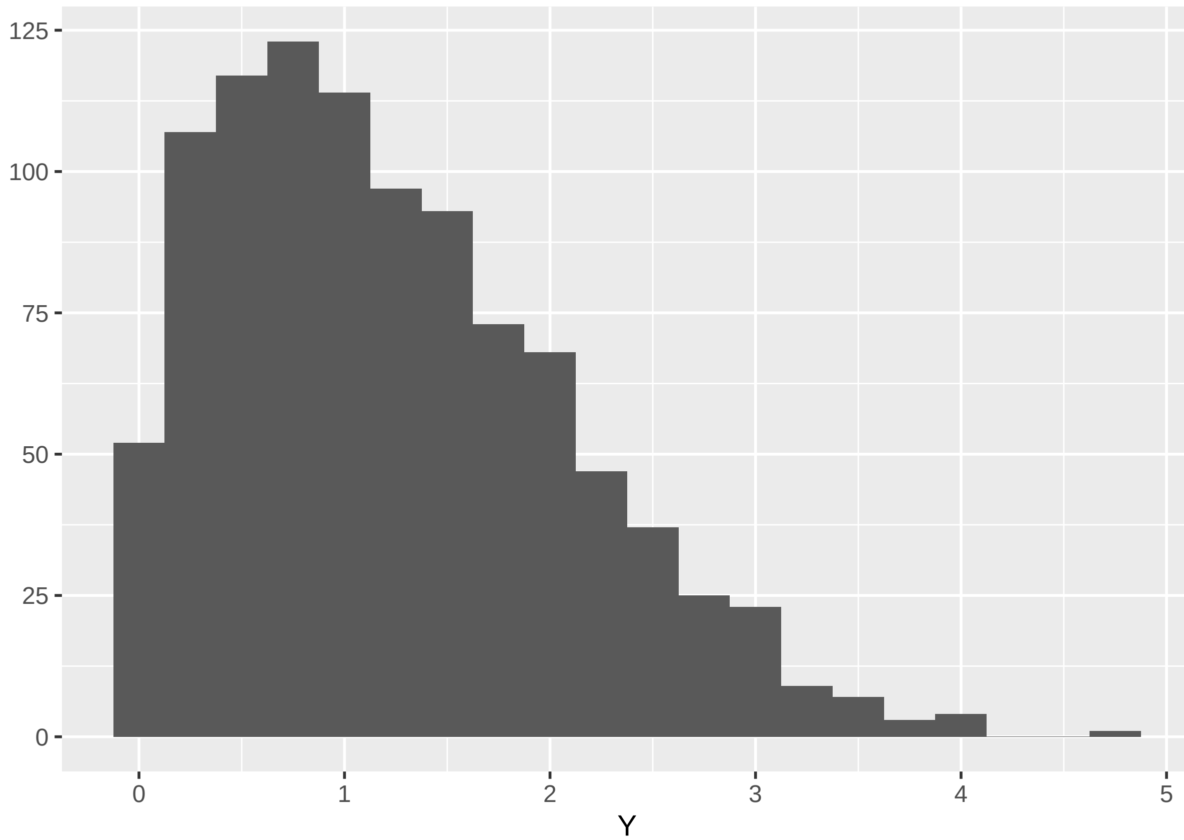
1/8/2022

Simulations

Nous observons un échantillons de variables aléatoires de loi, $Y_1, ..., Y_n$ avec

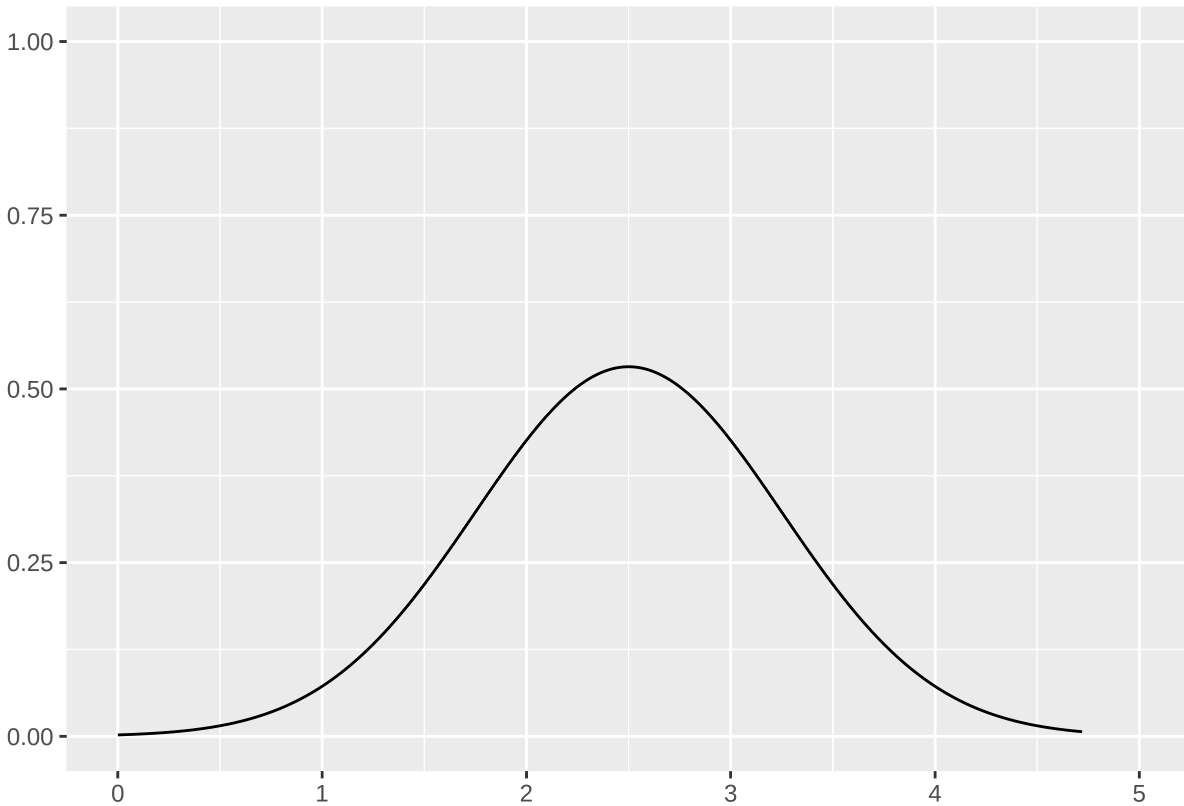
$$Y_i = X_i U_i, \forall i = 1, ..., n, U_i \sim \mathcal{U}(0,1), X_i \sim \mathcal{N}(2.5, 0.75)$$

```r
rm(list=ls())
library(ggplot2)

set.seed(116)
n <- 1000
X <- rnorm(n, 2.5, 0.75)
U <- runif(n)
Y <- X * U
qplot(Y, geom = "histogram", binwidth = 0.25)
```



On cherche donc à estimer la densité représenté par la fonction ci-dessous

```
f <- function(grid){
  X <- dnorm(grid, 2.5, 0.75)
  return(X)}
gridx <- seq(0,max(Y),length=500)
ggplot()+ aes(x=gridx) + geom_line(y=f(gridx),na.rm=T) + ylim(0,1)+ xlim(0, 5)+xlab('')+ ylab('')
```



Estimation de la densité

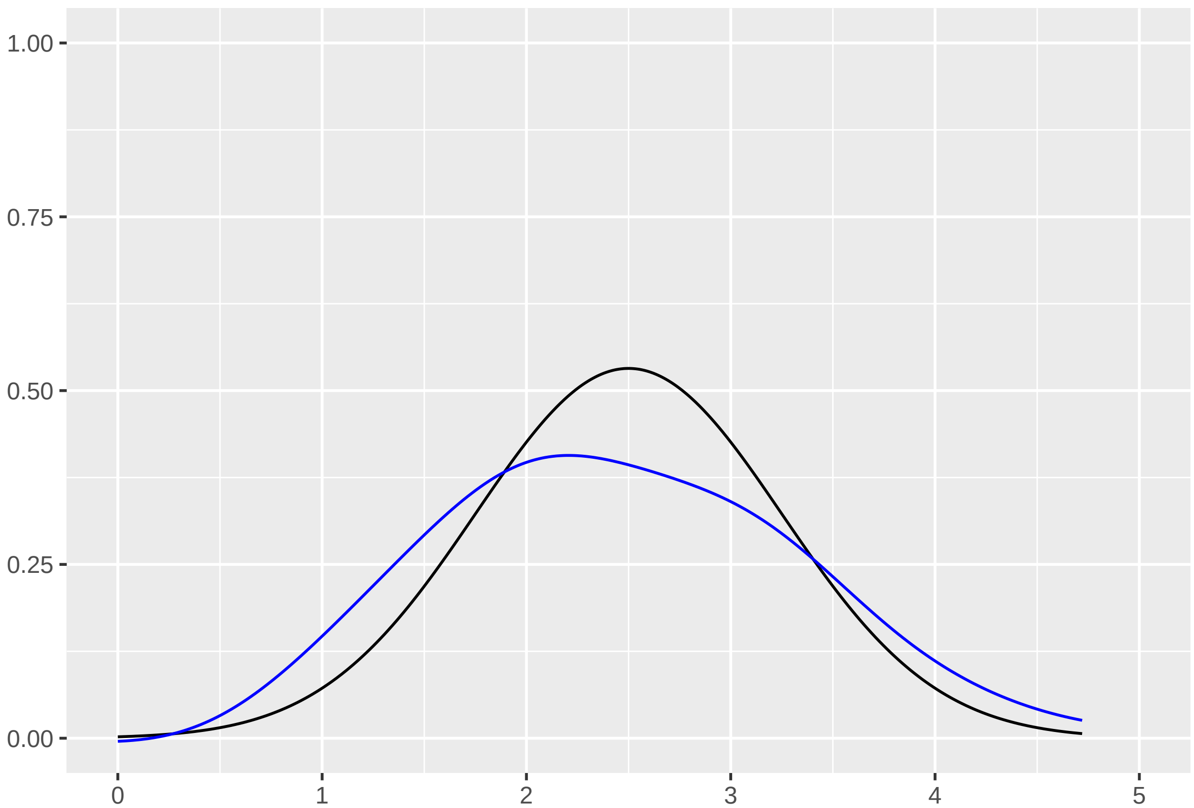
Nous implémentons par la suite l'estimateur à noyau de la fonction de densité du modèle de multiplicative censoring :

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i}{h}K_h'(Y_i - x) + K_h(Y_i - x)\right)$$

On utilise le noyau gaussien $k(u) = 1/(\sqrt{2\pi})\exp(-\frac{1}{2}u^2)$ de dérivée $k'(u) = -(u/\sqrt{2\pi})\exp(-\frac{1}{2}u^2)$.

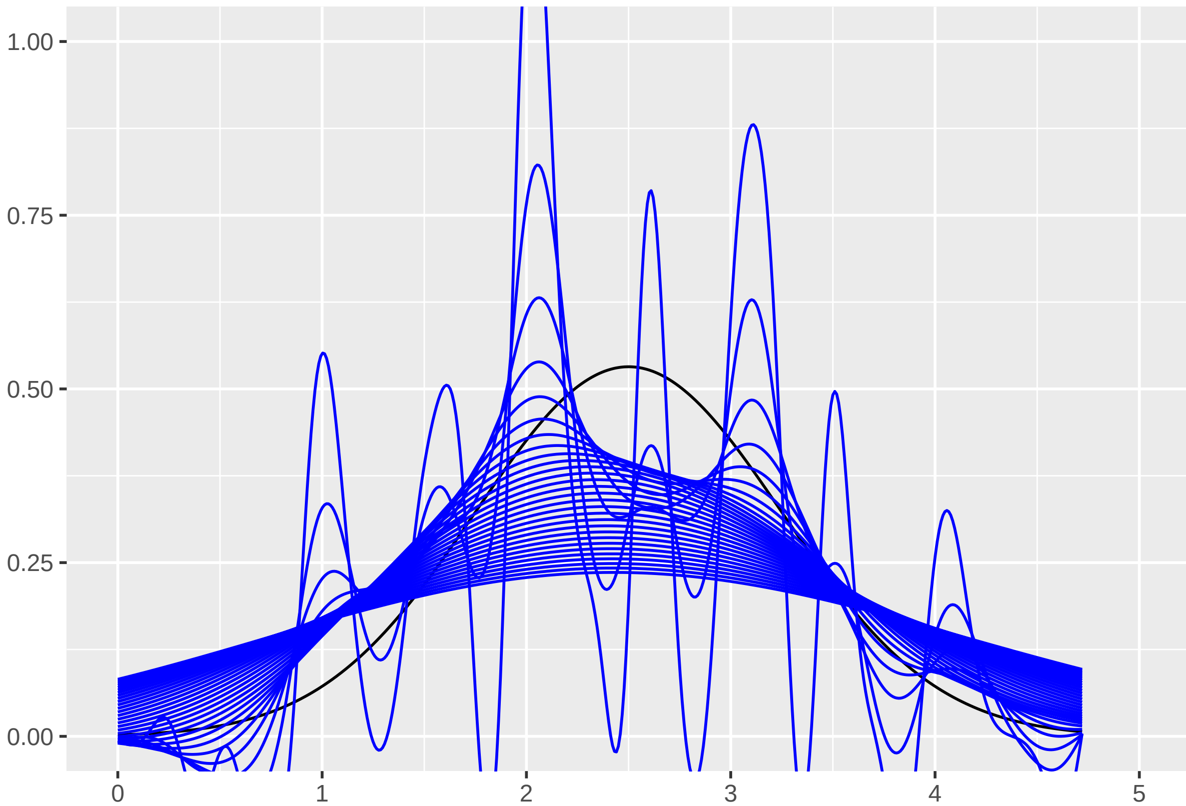
```
gaussian_kernel <- function(x){
  (1/sqrt(2*pi))*exp(-(1/2)*(x^2))
}
gaussian_kernel_prime <- function(x){
  - (x/sqrt(2*pi))*exp(-(1/2)*(x^2))
}

epanechnikov_kernel <- function(x){
  (3/4)*(1-x^2)*(abs(x)<=1)
}

epanechnikov_kernel_prime <- function(x){
  -(3/2)*x*(abs(x)<=1)
}



f_kernel <- function(x, K, Kprime, h, Y){
  fhat = (1/(n*h)) * sum(sapply(Y, function(Y_i) (Y_i/h)*Kprime((Y_i - x)/h) + K((Y_i - x)/h)))
  return(fhat = fhat)
```

```
}
N = length(X)
Xrep = matrix(X, nrow = length(gridx), ncol = N, byrow = TRUE)
gridrep = matrix(gridx, nrow = length(grid), ncol = N, byrow = FALSE)

h <- 0.5
K <- gaussian_kernel
Kprime <- gaussian_kernel_prime
estimf <- sapply(gridx, function(x) f_kernel(x, K, Kprime, h, Y))
ggplot() + aes(x=gridx)+geom_line(y=f(gridx),na.rm=T)+ ylim(0,1)+ xlim(0, 5)+xlab('')+ ylab('') +geom_li
```



Pour la collection de fenêtre utilisée par les auteurs de Comte & al. 2015 $\mathcal{H}_n = \{0.1 + 0.05k, k = 0, ..., 28\}$ on obtient :

```
gridh= sapply(0:28, function(k) 0.1+0.05*k)

collec <- matrix(sapply(gridh,function(h) sapply(gridx, function(x) f_kernel(x, K, Kprime, h, Y))), leng

plot1 <- ggplot()+ aes(x = gridx) + geom_line(y=f(gridx),na.rm=T)+ ylim(0,1)+ xlim(0, 5)+xlab('')+ ylab
for (i in 1:length(gridh)){
   plot1 <- plot1+ geom_line(y=collec[i,], col='blue', na.rm=T)
}
plot1
```



On a une majoration du risque intégré :

$$\mathbb{E}[\|\hat{f}_f - f\|^2] \leq \|f_h - f\|^2 + \|K\|^2/(nh) + \mathbb{E}[Y_1^2]\|K'\|^2/(nh^3)$$

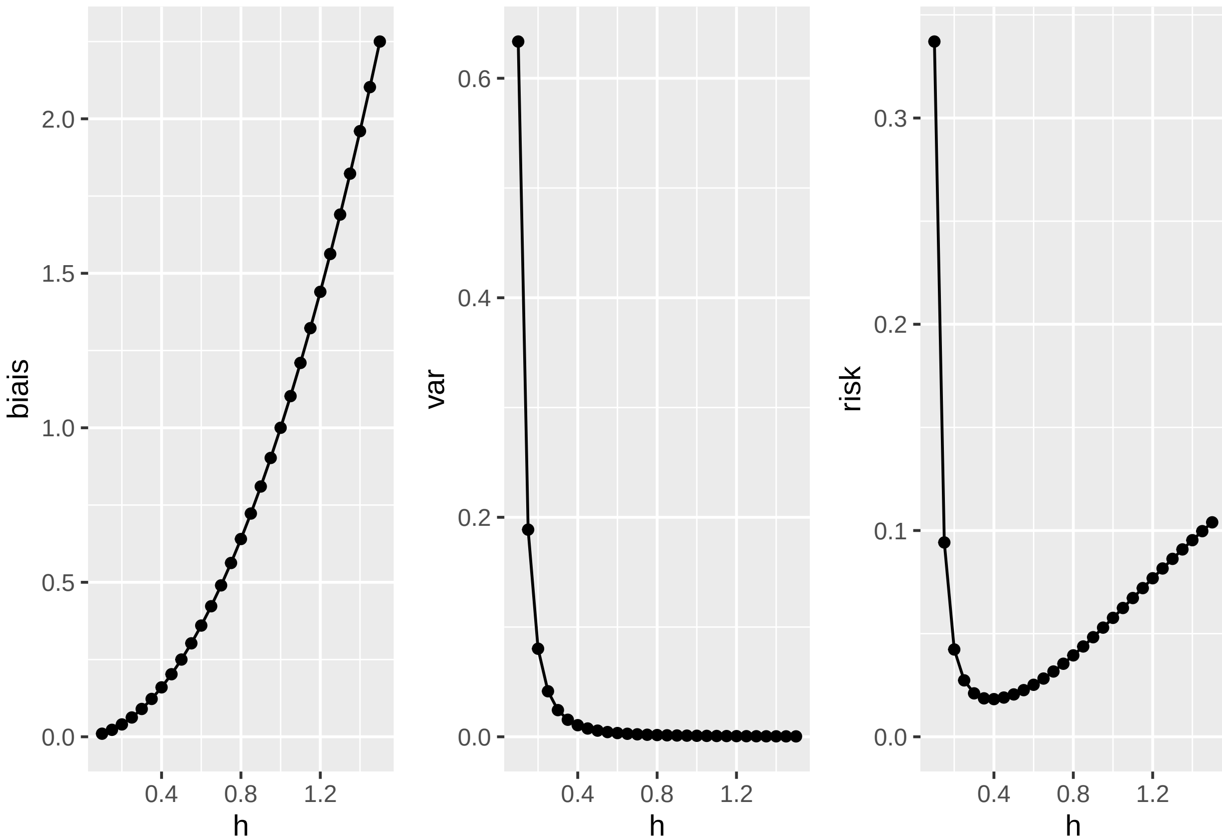
Pour un noyau gaussien on ales résultats suivant :

$$\|K\|^2 = \frac{1}{2\pi}\int \exp(-x^2)dx = \frac{1}{2\sqrt{\pi}}$$

$$\|K'\|^2 = \frac{1}{2\pi}\int x^2\exp(-x^2)dx = \frac{1}{2\sqrt{\pi}} = \frac{1}{\pi}\int_{\mathbb{R}_+} x^2\exp(-x^2)dx = \frac{1}{\pi}([\frac{x}{2}exp(-x^2)]_0^{+\infty} + \int_{\mathbb{R}_+}\exp(-x^2)dx) = \frac{1}{2\sqrt{\pi}}$$

On estime de plus $\mathbb{E}[Y_1^2]$ par la moyenne empirique $(1/n)\sum Y_i^2$. Avec une hypothèse faible sur la classe de f à savoir qu'elle est dans une classe de Nikol'ski de régularité au moins $\beta = 1$, on peut de plus majorer le biais au carré et on peut trouver de cette façon la fenêtre optimale h_{opt}.

```
risk <-  ((max(gridx)-min(gridx))/length(gridx))*apply((collec-matrix(f(grid=gridx), length(gridh), leng
var  <- (1/n) * (1/gridh)*(1/(2*sqrt(pi))) + (1/n) * (1/(gridh^3))*mean(Y^2)*(1/(2*sqrt(pi)))
biais <- gridh^2

library(gridExtra)

plot1 <- ggplot()+ aes(x=gridh)+geom_point(y= biais)+geom_line(y=biais)+ xlab('h')+ ylab('biais')  + yli
plot2 <- ggplot()+ aes(x=gridh)+geom_point(y= var)+geom_line(y= var)+ xlab('h')+ ylab('var') + ylim(0,ma
plot3 <- ggplot()+ aes(x=gridh)+geom_point(y= risk)+geom_line(y= risk)+ xlab('h')+ ylab('risk')  + ylim

grid.arrange(plot1, plot2, plot3, ncol=3, nrow = 1)
```

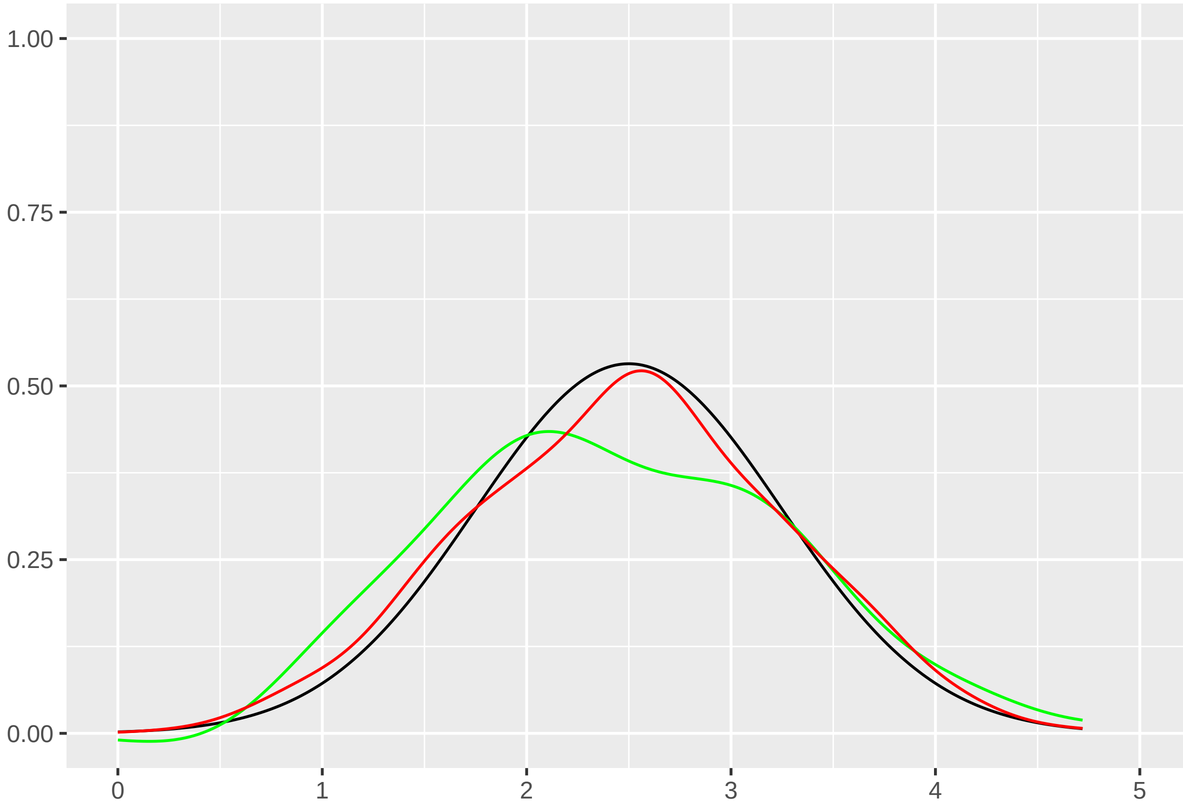



```r
horacle <- gridh[which.min(risk)]
paste('horacle=',horacle)
```

```
## [1] "horacle= 0.4"
```

On voit ici la comparaison entre l'estimateur par noyau avec h_{oracle} qui minimise le risque (vert), l'estimation par validation croisée en utilisant directement les X_i (rouge) et la fonction à estimer (noir).

```r
K <- gaussian_kernel
Kprime <- gaussian_kernel_prime
estimf <- sapply(gridx, function(x) f_kernel(x, K, Kprime, horacle, Y))
plot1 <- ggplot()+ aes(x=gridx)+geom_line(y=f(gridx),na.rm=T)+ ylim(0,1)+ xlim(0, 5)+xlab('')+ ylab('')

plot2 <- ggplot()+ aes(x=gridx)+geom_line(y=f(gridx), na.rm=T)+ ylim(0,1)+ xlim(0, 7.5)+xlab('')+ ylab(
fucv <- density(X, from=min(gridx), to=max(gridx), cut=diff(gridx), n=length(gridx), bw="ucv" )
plot1 +  geom_line(y= fucv$y, colour='red',na.rm=T)
```


1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5