Homework4

Ejercicio 2. Densidad gaussiana.

Se pide

$$\mathbb{P}[X \leq 15]$$

. Se hará de la siguiente manera, con el estimador de Kernel Rosenblatt, con K_0 un kernel gaussino.

$$\mathbb{P}[X \leq 15] = \int_{-\infty}^{15} \hat{f}_n^{K_0}(x)dx = \frac{1}{\sqrt{9}} \int_{-\infty}^{15} \sum_{i=1}^{9} \varphi(\frac{x_i - x}{\sqrt{1/3}})dx = \frac{1}{9} \sum_{i=1}^{9} \Phi(\frac{15 - x_i}{\sqrt{1/3}})$$

```
#data
X <- c(12,13,12,15,15,16,19,20,25)
n <- length(X)
p <- 15
hn <- sqrt(3/n)

#estimate P(X < 15)
s <- 0
for (xi in X){s <- s + pnorm((p-xi)/hn)}
s <- 1/n * s
```

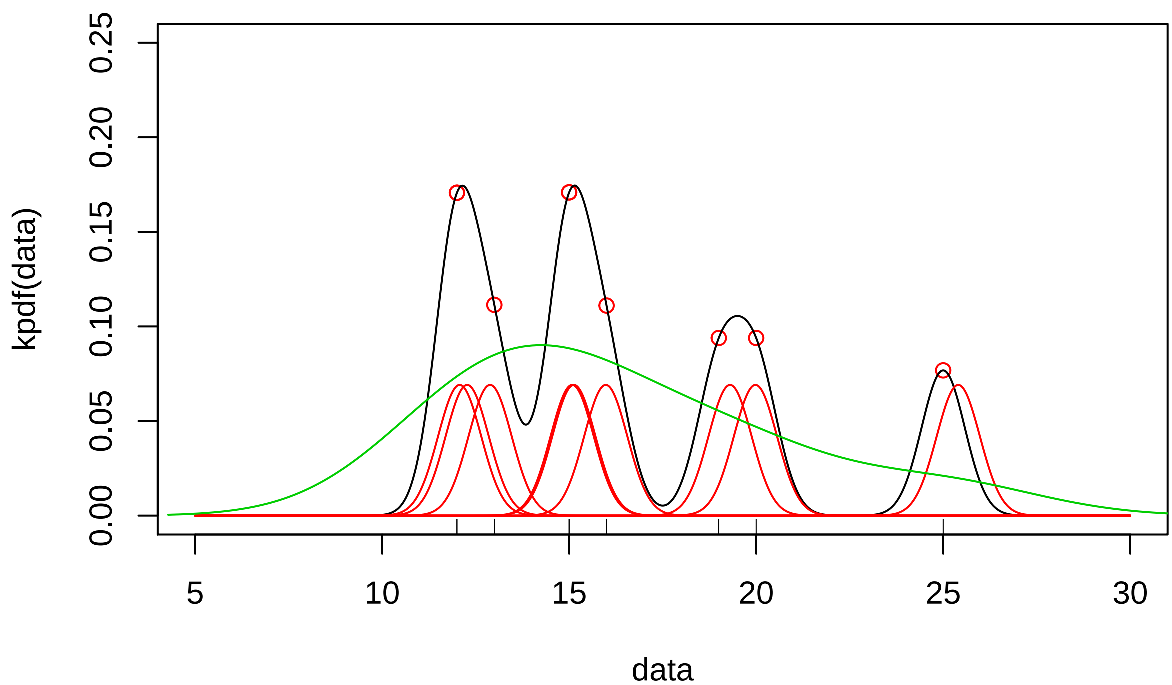
Por tanto,

$$\mathbb{P}[X \leq 15] \approx$$

0.4490407

```
#seed
set.seed(42)

#density estimator
#data
data <- c(12,13,12,15,15,16,19,20,25)
n <- length(X)
p <- 15
hn <- sqrt(3/n)

# Kernel
Kernel <- function(x) {mean(dnorm((x - data)/ hn)/hn )}    #kernel Density
kpdf <- function(x) sapply(x, Kernel)   # Elementwise application


# estimate kernel
x <- seq(5, 30, length = 1000)

#plot
#total prob
plot(data, kpdf(data), col = 2, ylim = c(0,.25), xlim = c(5,30))

#plot all
```

```r
lines(x, kpdf(x))

#rug
rug(data)

#individual kernels
i <- 1
eps <- rnorm(n = n, sd = 0.2)
for(xi in data){
  lines(x, dnorm(x, mean = xi + eps[i], sd = hn)*.1, col = 2)
  i <- i + 1
}

#density with R
lines(density(data, kernel = 'gaussian', bw = 'SJ'), xlim = c(5,30), ylim = c(0,.20), col = 3)
```



La probabilidad estimada de manera numérica está dada por:

```r
# estimate probability
integrate(kpdf, - Inf, 15)
```

```
## 0.4490407 with absolute error < 8.6e-06
```

NOTA

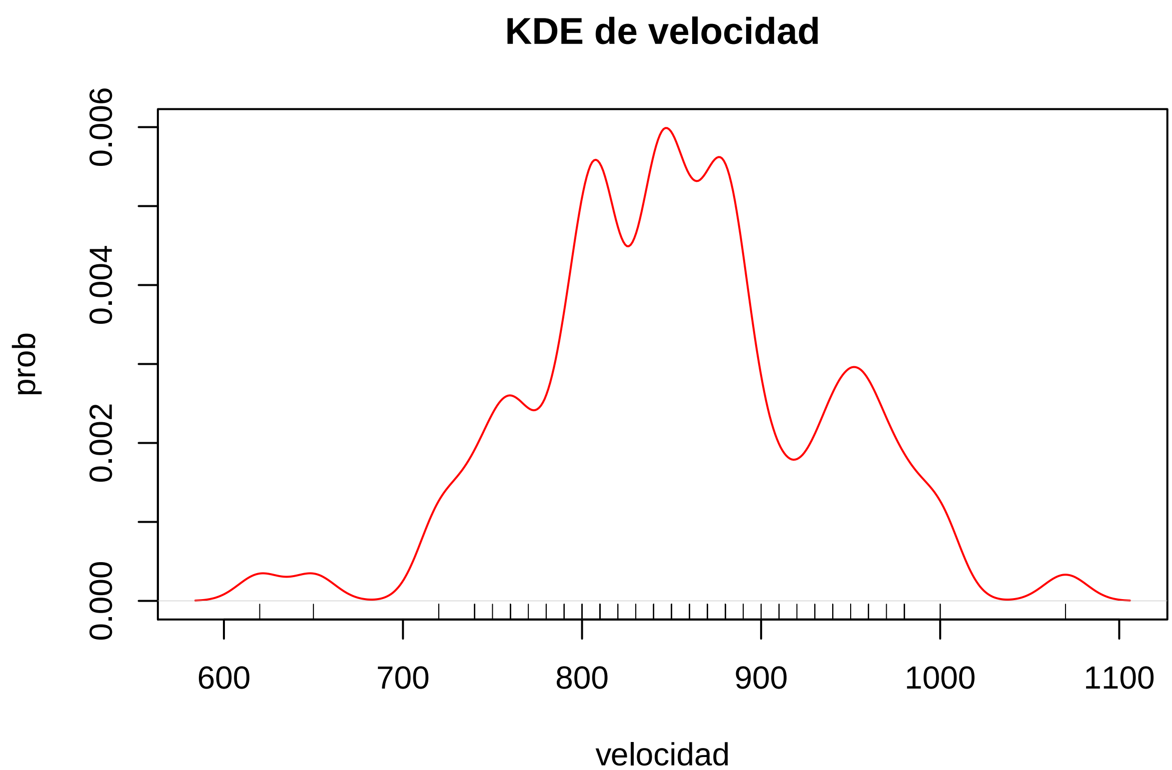
1. A cambios de h_n , las probabilidades cambian.
2. La estimación de la probabilidad numérica coincide con la teórica.

##Ejercicio 4

```
data(Michelson)
head(Michelson)
```

```
##   velocity
## 1      850
## 2      740
## 3      900
## 4     1070
## 5      930
## 6      850
```

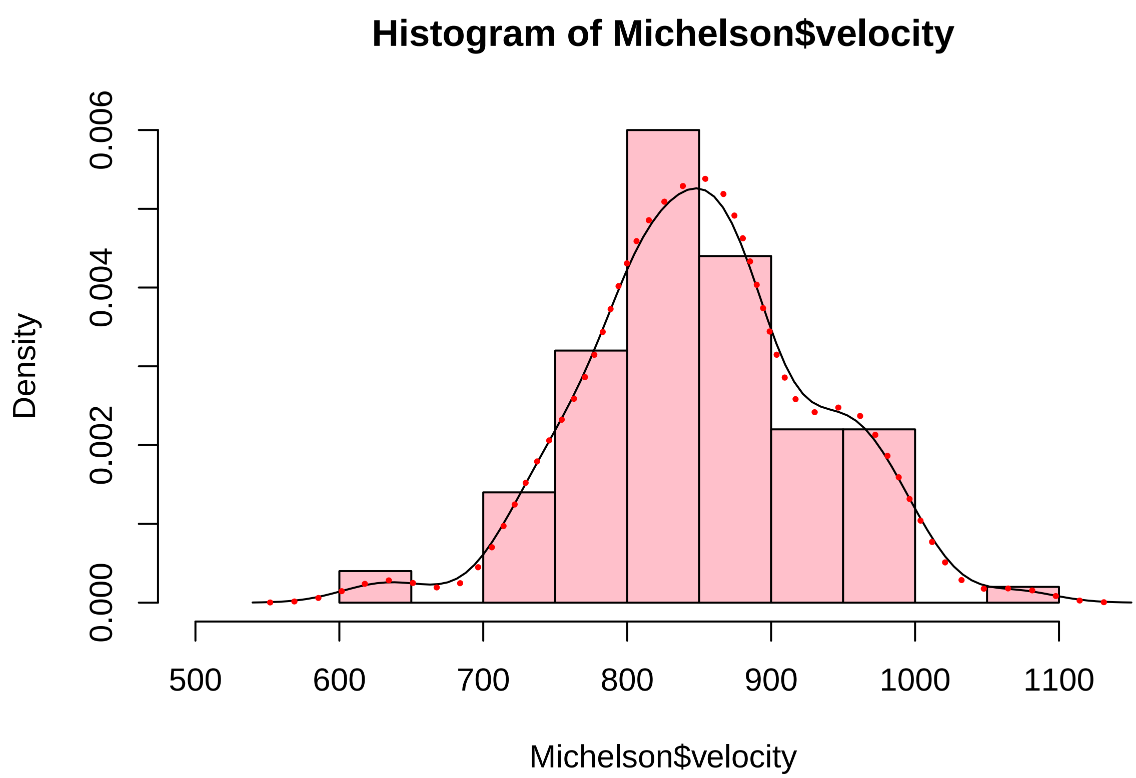
```
plot(density(Michelson$velocity,bw = 12),col = "red", lwd =1, main = "KDE de velocidad",ylab="prob",xlab
rug(Michelson$velocity)
```




```
hist(Michelson$velocity, breaks=15, xlim = c(500,1150), ylim = c(0,0.006), col ="pink", prob = T)
lines(density(Michelson$velocity, bw = "nrd", n = 100)) # factor normal (1.06)
bandwidth.nrd(Michelson$velocity) # ancho de banda cuadruplicado.
```

```
## [1] 107.073
```

```r
lines(density(Michelson$velocity, bw = "nrd0", n = 100), lty=3, col = "red", lwd = 3) #factor con (0.9)
```



##Ejercicio 6

```r
set.seed(042020)
u <- runif(100)
x <- sort(u)
y <- x^2 + 0.1*rnorm(100)
```

```r
splinefun(x, y,  method = "fmm", ties = mean)
```

```
## function (x, deriv = 0L)
## {
##     deriv <- as.integer(deriv)
##     if (deriv < 0L || deriv > 3L)
##         stop("'deriv' must be between 0 and 3")
##     if (deriv > 0L) {
##         z0 <- double(z$n)
##         z[c("y", "b", "c")] <- switch(deriv, list(y = z$b, b = 2 *
##             z$c, c = 3 * z$d), list(y = 2 * z$c, b = 6 * z$d,
##             c = z0), list(y = 6 * z$d, b = z0, c = z0))
##         z[["d"]] <- z0
##     }
##     res <- .splinefun(x, z)
##     if (deriv > 0 && z$method == 2 && any(ind <- x <= z$x[1L]))
```

```
##         res[ind] <- ifelse(deriv == 1, z$y[1L], 0)
##     res
## }
## <bytecode: 0x00000000193e4718>
## <environment: 0x00000000193e9400>
```

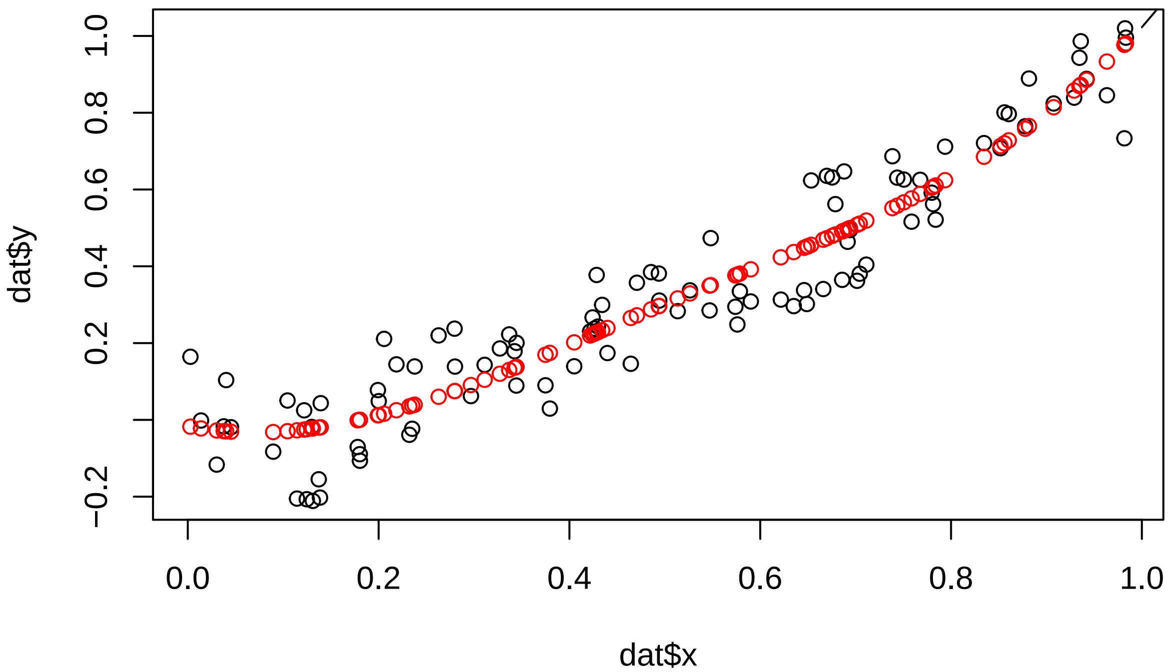
```r
dat <- data.frame(x, y)
FIT <- lm( y ~ bs(x, df = 4 ), data =dat)
summary(FIT)
```

```
##
## Call:
## lm(formula = y ~ bs(x, df = 4), data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24347 -0.08010  0.01012  0.07315  0.19487
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.01769    0.04963  -0.356    0.722
## bs(x, df = 4)1 -0.07530    0.09914  -0.760    0.449
## bs(x, df = 4)2  0.36535    0.07814   4.675 9.68e-06 ***
## bs(x, df = 4)3  0.59868    0.08641   6.929 5.05e-10 ***
## bs(x, df = 4)4  0.99842    0.06229  16.029  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.102 on 95 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8929
## F-statistic: 207.4 on 4 and 95 DF,  p-value: < 2.2e-16
```

```r
MAT <- bs(dat$x, df = 4)
X <- dat[[1]]
Ypred <- MAT[, 1] * FIT$coefficients[2] +
  MAT[, 2] * FIT$coefficients[3] +
  MAT[, 3] * FIT$coefficients[4] +
  MAT[, 4] * FIT$coefficients[5] + FIT$coefficients[1]

plot(dat$x, dat$y)
Pts <- seq(1,10, length.out = 100)
lines(Pts, predict(FIT, newdata = list(x = Pts)))
```

```
## Warning in bs(x, degree = 3L, knots = c(`50%` = 0.493954221135937),
## Boundary.knots = c(0.00274603301659226, : some 'x' values beyond boundary
## knots may cause ill-conditioned bases
```

```r
points(X, Ypred, col = "red")
```


dat$y
1.0
0.8
0.6
0.4
0.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
dat$x