

**Maestría en Computo Estadístico**  
**Inferencia Estadística**  
**Tarea 2**

20 de septiembre de 2020

*Enrique Santibáñez Cortés*

Repositorio de Git: [Tarea 2, IE](#).

4. El siguiente conjuntos de datos contiene mediciones del diámetro de un agave, medido en decímetros, en distintas localizaciones no cercanas.

23.37	21.87	24.41	21.27	23.33	15.20	24.21	27.52	15.48	27.19
25.05	20.40	21.05	28.83	22.90	18.00	17.55	25.92	23.64	28.96
23.02	17.32	30.74	26.73	17.22	22.81	20.78	23.17	21.60	22.37

- a) Escriba una función en R que calcule la función de distribución empírica para un conjunto de datos dado  $D$ . La función debe tomar como parámetros al valor  $x$  donde se evalúa y al conjunto de datos  $D$ . Utilizando esta función grafique la función de distribución empírica asociada al conjunto de datos de agave. Ponga atención a los puntos de discontinuidad. ¿Qué observa? **Nota:** Escriba la función mediante el algoritmo descrito en las notas de la clase; para este ejercicio no vale usar la funciones implementadas en R que hacen lo pedido.

### RESPUESTA

Sea  $X_1, \dots, X_n \sim F$ , la función de densidad empírica  $\hat{F}$  es la función de distribución acumulada que asigna masa  $1/n$  en cada punto  $X_i$ . Formalmente,

$$\hat{F}(x) = \frac{\sum_{i=1}^n 1_{X_i \leq x}}{n},$$

donde

$$1_{X_i \leq x} = \begin{cases} 1 & \text{si } X_i \leq x \\ 0 & \text{si } X_i > x \end{cases}$$

Utilizando la definición anterior construimos la función en R:

```
fde <- function(x, D){  
  n <- length(D)  
  if(length(x)>1){  
    fde_x <- c()  
    for(i in 1:length(x)){  
      fde_x[i] <- sum(D<=x[i])/n  
    }  
  }  
  else{fde_x <- sum(D<=x)/n}  
  fde_x  
}
```

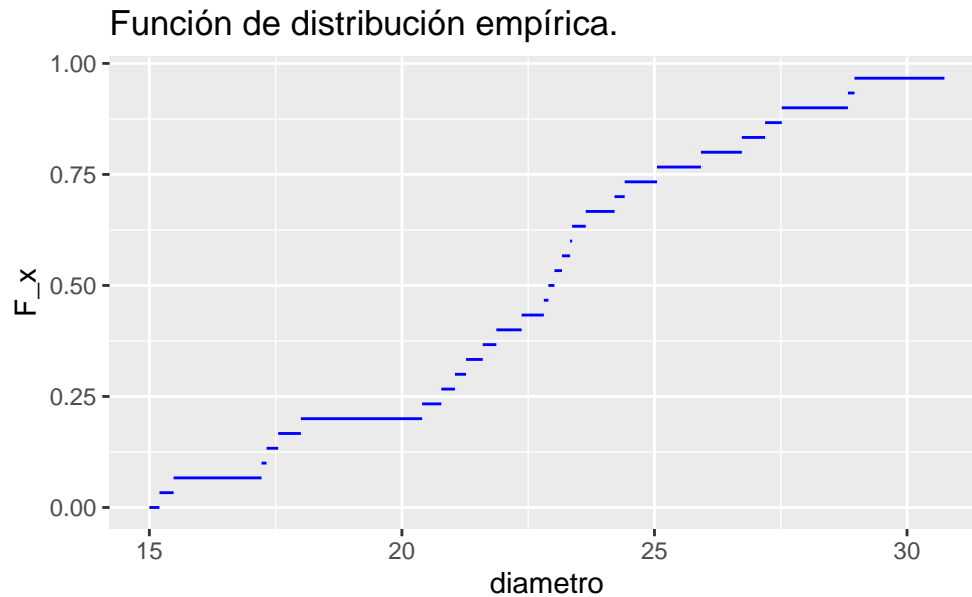
Procedemos a calcular con la función anterior la fde en los puntos de discontinuidad, para ello primero ingresemos los datos:

```
library(tidyverse)  
df_agave <- data.frame(diametro=c(23.37, 21.87, 24.41, 21.27, 23.33, 15.20, 24.21, 27.52, 15.48,  
25.05, 20.40, 21.05, 28.83, 22.90, 18.00, 17.55, 25.92, 23.64, 28.96, 23.02, 17.32,  
30.74, 26.73, 17.22, 22.81, 20.78, 23.17, 21.60, 22.37))  
  
df_agave <- df_agave %>% arrange(diametro)
```

Los puntos de discontinuidad serían cuando cambia fde, es decir, cuando se evalúa en un  $x + \epsilon$  tal que  $\hat{F}(x) \neq \hat{F}(x + \epsilon)$ , con los datos podemos decir que los puntos de discontinuidad son los estadísticos de orden  $x_{(i)}$ .

```
df_agave_fde <- df_agave %>%
  mutate(F_x=fde(x=diametro, D=df_agave$diametro),
         xend=lag(x=diametro, n = 1, default = 15))

ggplot(df_agave_fde)+
  geom_segment(aes(x=diametro, xend=xend, y=F_x, yend=F_x), col="blue")+
  labs(title = "Función de distribución empírica.")
```



- b) Usando la desigualdad de Dvoretzky-Kiefer-Wolfowitz, escriba una función en R que calcule y grafique una región de confianza para la función de distribución empírica. La función debe tomar como parámetros al conjunto de datos que se usan para contruir la función de distribución empírica.

## RESPUESTA

Enunciemos la desigualdad de Dvoretzky-Kiefer-Wolfowitz:

**Teorema: 1** (Desigualdad de Dvoretzky-Kiefer-Wolfowitz) Sean  $X_1, \dots, X_n \sim F$  independientes. Entonces, para todo  $\epsilon > 0$ ,

$$\mathbb{P} \left( \sup_x |\hat{F}_n(x) - F(x)| > \epsilon \right) \leq 2e^{-2n\epsilon^2}.$$

A partir de lo anterior, podemos construir una región de confianza para la distribución empírica. Sea:

$$L(x) = \max \{ \hat{F}_n - \epsilon_n, 0 \},$$

$$U(x) = \min \{ \hat{F}_n + \epsilon_n, 1 \}$$

donde

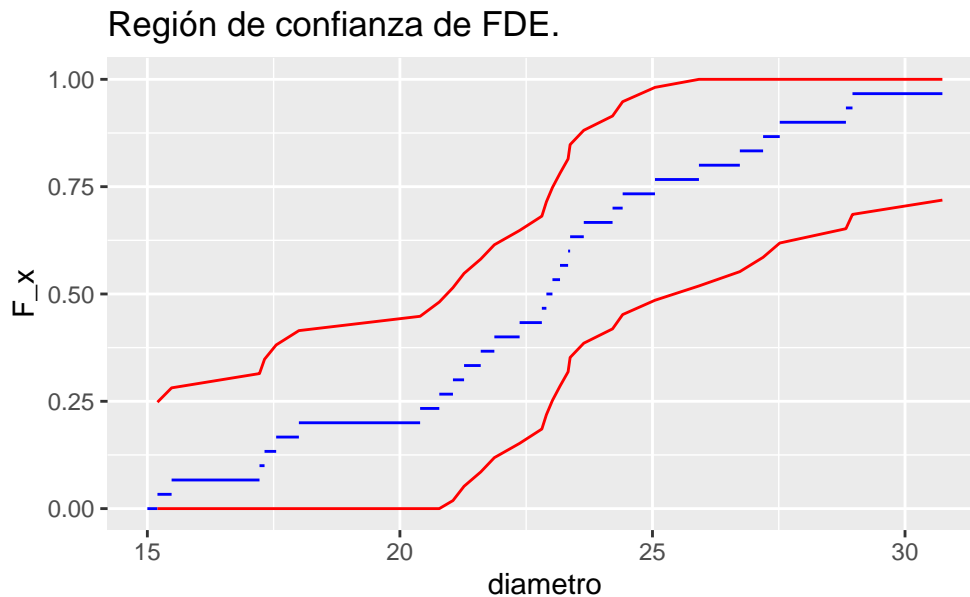
$$\epsilon_n = \sqrt{\frac{1}{2n} \log \left( \frac{2}{\alpha} \right)}.$$

Puede verse que  $F(x)$

$$\mathbb{P}(L(x) \leq F(x) \leq U(x) \text{ para todo } x) \geq 1 - \alpha.$$

Entonces, una región de confianza con  $\alpha = 0.05$  para la distribución empírica se puede programar de la siguiente forma:

```
graphisc_fde <- function(df){  
  alpha <- 0.05  
  df <- df %>%  
    mutate(F_x=fde(x=diametro, D=df_agave$diametro),  
           xend=lag(x=diametro, n=1, default = 15),  
           L_x=F_x-sqrt((1/(2*30))*log(2/alpha)),  
           L_x=if_else(L_x<0,0, L_x),  
           U_x=F_x+sqrt((1/(2*30))*log(2/alpha)),  
           U_x=if_else(U_x>1,1, U_x))%>%  
    gather(key=".", value="y", c(4,5))  
  df  
}  
df_agave_fde_bandas <- graphisc_fde(df_agave)  
  
ggplot(df_agave_fde)+  
  geom_segment(aes(x=diametro, xend=xend, y=F_x, yend=F_x), col="blue")+  
  labs(title = "Región de confianza de FDE.")+  
  geom_line(data=df_agave_fde_bandas, aes(x=diametro, y=y,group=.), col="red")
```



- c) Escriba una función en R que determine la gráfica Q-Q normal de un conjunto de datos. La función debe tomar como parámetro al conjunto de datos y deberá graficar contra el percentil estandarizado de la normal. Para poder comparar el ajuste más claramente, la función además deberá ajustar en rojo a la recta  $sx + \bar{x}$  ( $s$  =desviación estándar muestral y  $x$ =media muestral). Usando esta función, determine la gráfica Q-Q normal. ¿Qué observa? **Nota:** La misma del inciso a).

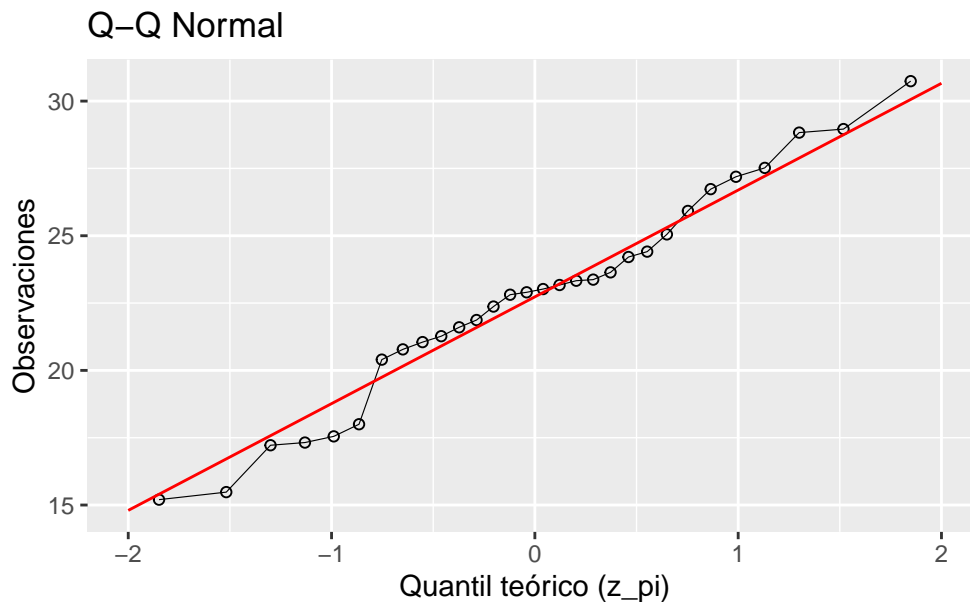
**RESPUESTA**

```
sd <- sqrt(var(df_agave$diametro))
mu <- mean(df_agave$diametro)
recta <- data.frame(x=c(-2,2), y=c(-sd*2+mu, sd*2+mu))

df_agave$i <- 1:nrow(df_agave)

df_agave <- df_agave %>%
  mutate(pi= i/31,
         z_pi=qnorm(pi))

ggplot(df_agave, aes(x=z_pi, y=diametro))+
  geom_point(shape=1)+
  geom_line(size=0.2)+
  geom_line(data=recta, aes(x=x, y=y), col="red")+
  labs(title = "Q-Q Normal", x="Quantil teórico (z_pi)", y="Observaciones")
```



- d) Escriba una función en R que determine el gráfico de probabilidad normal. La función debe tomar como parámetro al conjunto de datos. ¿Qué observa? Nota: La misma del inciso a).

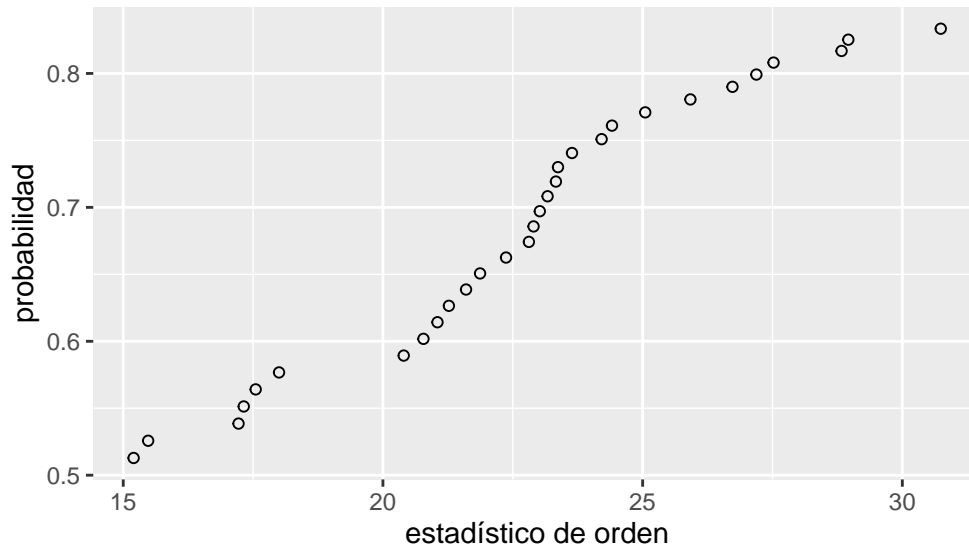
## RESPUESTA

Utilizando:

```
df_agave <- df_agave %>%
  mutate(z_pi_in = pnorm(pi))

ggplot(df_agave, aes(y=z_pi_in, x=diametro))+
  geom_point(shape=1)+
  labs(title = "Grafico de probabilidad normal", y="probabilidad", x="estadístico de orden")
```

Grafico de probabilidad normal



e) ¿Los datos anteriores se distribuyen normalmente? Argumente.

5. En este ejercicio repasará la estimación de densidades.

a) Escriba una función en R que estime una densidad por el método de kernels. La función deberá recibir al punto  $x$  donde se evalúa al estimador, al parámetro de suavidad  $h$ , al kernel que se utilizará en la estimación y al conjunto de datos.

## RESPUESTA

Para el caso univariado, el  $KDE$  está dado por

$$\hat{f} = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right), \quad x \in \mathbb{R}, h > 0.$$

$K$  es la función Kernel, y  $h$  es el ancho de banda que determina la suavidad de la estimación. Procedemos a definir las funciones Kernel's, solo consideraremos las siguientes: gaussian, epanechnikov, rectangular, triangular, consine.

```
# Definimos las funciones kernel:
kernel_gaussian <- function(x,x_i,h){
  dnorm((x-x_i)/h)
}

kernel_rectangular <- function(x,x_i,h){
  dunif((x_i - x)/h, min = -1, max = 1)
}

kernel_triangular <- function(x, x_i, h){
  u = (x_i-x)/h
  (1-abs(u))*(abs(u) <= 1)
}

kernel_epanechnikov <- function(x, x_i, h){
  u = (x_i - x)/h
  3/4*(1-u^2)*(abs(u) <= 1)
```

```

}

kernel_cosine <- function(x, x_i, h){
  u = (x_i - x)/h
  (1+cos(pi*u))/2*(abs(u) <= 1)
}

```

Una vez definido lo anterior procedemos a definir la función para estimar la densidad por el método de kernels ([https://github.com/JonasMoss/kdensity/blob/master/R/builtin\\_kernels.R](https://github.com/JonasMoss/kdensity/blob/master/R/builtin_kernels.R)):

```

kde <- function(x, h, kernel, D){
  f_hat <- 0
  n <- length(D)
  for (i in 1:n){
    if (kernel=="gaussian"){
      f_hat <- f_hat + kernel_gaussian(x, D[i], h)
    }
    else if(kernel=="epanechnikov"){
      f_hat <- f_hat + kernel_epanechnikov(x, D[i], h)
    }
    else if(kernel=="rectangular"){
      f_hat <- f_hat + kernel_rectangular(x, D[i], h)
    }
    else if(kernel=="triangular"){
      f_hat <- f_hat + kernel_triangular(x, D[i], h)
    }
    else if(kernel=="consine"){
      f_hat <- f_hat + kernel_cosine(x, D[i], h)
    }
    else{
      print("Kernel incorrecto")
      break()
    }
  }
  f_hat/(n*h)
}

```

- b) Cargue en R al archivo “Tratamiento.csv”, el cual contiene la duración de los períodos de tratamiento (en días) de los pacientes de control en un estudio de suicidio. Utilice la función del inciso anterior para estimar la densidad del conjunto de datos para  $h = 20, 30, 60$ . Grafique las densidades estimadas. ¿Cuál es el mejor valor para  $h$ ? Argumente.

## RESPUESTA

Cargamos los datos:

```
tratamiento <- read.csv("Tratamiento.csv")
```

Graficamos la densidad estimada con el método de Kernel, con diferentes anchos de bandas.

```

x <- seq(-50, 800, 1)
fd_tratamiento <- data.frame(x=rep(x, 3),
                             fd=c(kde(x, 20, "gaussian", tratamiento$X1),

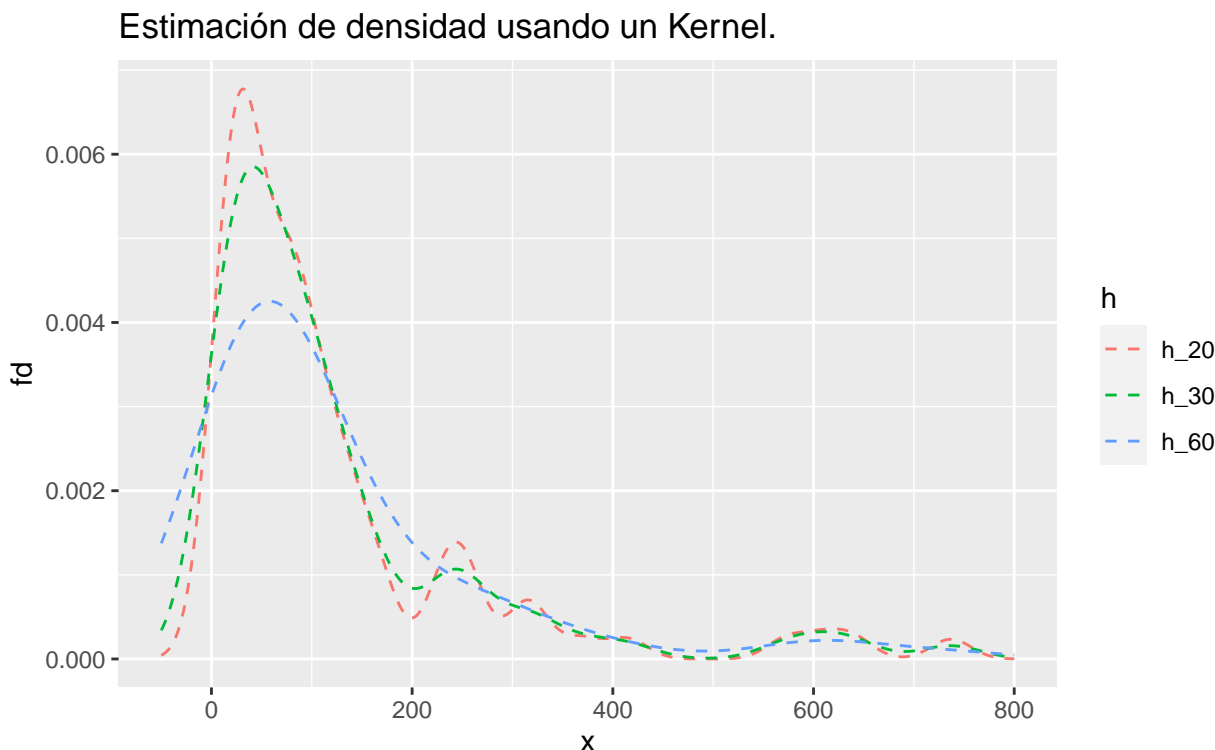
```

```

kde(x, 30, "gaussian", tratamiento$X1),
kde(x, 60, "gaussian", tratamiento$X1)),
h=c(rep("h_20", 851), rep("h_30", 851), rep("h_60", 851)))

ggplot(fd_tratamiento, aes(x=x, y=fd)) +
  geom_line(aes(colour=h), linetype=2) +
  labs(title = "Estimación de densidad usando un Kernel.")

```



Desde mi perspectiva el mejor valor de ancho de banda  $h$  es igual a 30, comparado con  $h = 20$  sobreestima a los datos eso se observa en las pequeñas montañas que se aprecian cuando  $x \approx 230, 250$ ; y comparandolo con  $h = 60$  observamos que se subestima a los datos.

- c) En el contexto de la estimación de densidades, escriba una función en R que determine el ancho de banda que optimiza al ISE. Grafique la densidad con ancho de banda óptimo para el conjunto de datos de “Tratamiento.csv”.

## RESPUESTA

Ocuparemos el método de Least squares cross validation (LSCV). Tenemos que

$$\begin{aligned}
 ISE(h) &= \int_{-\infty}^{\infty} (\hat{f}(x) - f(x))^2 dx \\
 &= \int_{-\infty}^{\infty} \hat{f}^2(x) dx - 2 \int_{-\infty}^{\infty} \hat{f}(x) f(x) dx + \int_{-\infty}^{\infty} f^2(x) dx
 \end{aligned}$$

Si observamos, la última integral de la expresión no depende del estimador  $\hat{f}(x)$  (es constante), por lo que la elección del ancho de banda (en el sentido de minimizar el ISE) corresponderá a la elección de  $h$  que minimiza la función:

$$R(\hat{f}) = \int_{-\infty}^{\infty} \hat{f}^2(x) dx - 2 \int_{-\infty}^{\infty} \hat{f}(x) f(x) dx + \int_{-\infty}^{\infty} f^2(x) dx.$$

La segunda parte usando un estimador de la densidad leave-one-out,  $\hat{f}_{-i}(x)$  es:

$$\hat{f}_{-i}(x) = \frac{1}{n-1} \sum_{j \neq i} K(x, x_j)$$

el cual es un estimador de la función de densidad usando todas los datos exepcto  $x_i$ . El resultado del criterio LSCV es:

$$LSCV(h) = \int_{-\infty}^{\infty} \hat{f}^2(x) dx - \frac{2}{n} \sum_i \hat{f}_{-i}(x_i).$$

El parametro  $h$  optimo es el valor que miniza la función  $LSCV(h)$ . (Weglarczyk 2018)

Lo anterior igual puede ser visto como (Härdle et al. 1991)

$$LSCV(h) = \frac{R(K)}{nh} + \frac{1}{n(n-1)h} \sum_{i=1}^n \sum_{\substack{j=1, \\ j \neq i}}^n (K * K - 2K) \left( \frac{X_j - X_i}{h} \right).$$

donde para un kernel normal

$$R(K) = \frac{1}{2\sqrt{\pi}}$$

$$K * K(x) = \frac{1}{2\pi} e^{-\frac{1}{2} \left( \frac{x}{\sqrt{2}} \right)^2}.$$

Utilizando lo anterior podemos buscar  $h$  que hace optimo  $LSCV(h)$ . Para hacerlo eficiente creamos una matriz de  $n \times n$  donde el elemento (i,j) es igual a  $K \left( \frac{X_i - X_j}{h} \right)$  para  $i \neq j$ , y 0 si  $i = j$ .

```
completa <- function(x, kernel="gaussian"){
  n <- length(x)
  upper=2*70
  lower=0.1*70
  tol=0.1 * lower

  LSCV_h <- function(h){
    D <- dnorm(outer(x,x,"-")/h)
    diag(D) <- 0
    D <- (1 / ((n-1)*h)) * colSums(D)
    D1 <- mean(D)
    D2 <- dnorm(outer(x,x,"-")/h,mean=0,sd=sqrt(2))
    diag(D2) <- 0
    D3 <- (1/((n-1)*h)) * colSums(D2)
    D4 <- mean(D3)
    (1/(n*h)) * (1/(2*sqrt(pi))) + D4 - 2*D1
  }

  obj <- optimize(LSCV_h , c(lower, upper), tol=tol)
  obj
}

completa(tratamiento$X1)
```

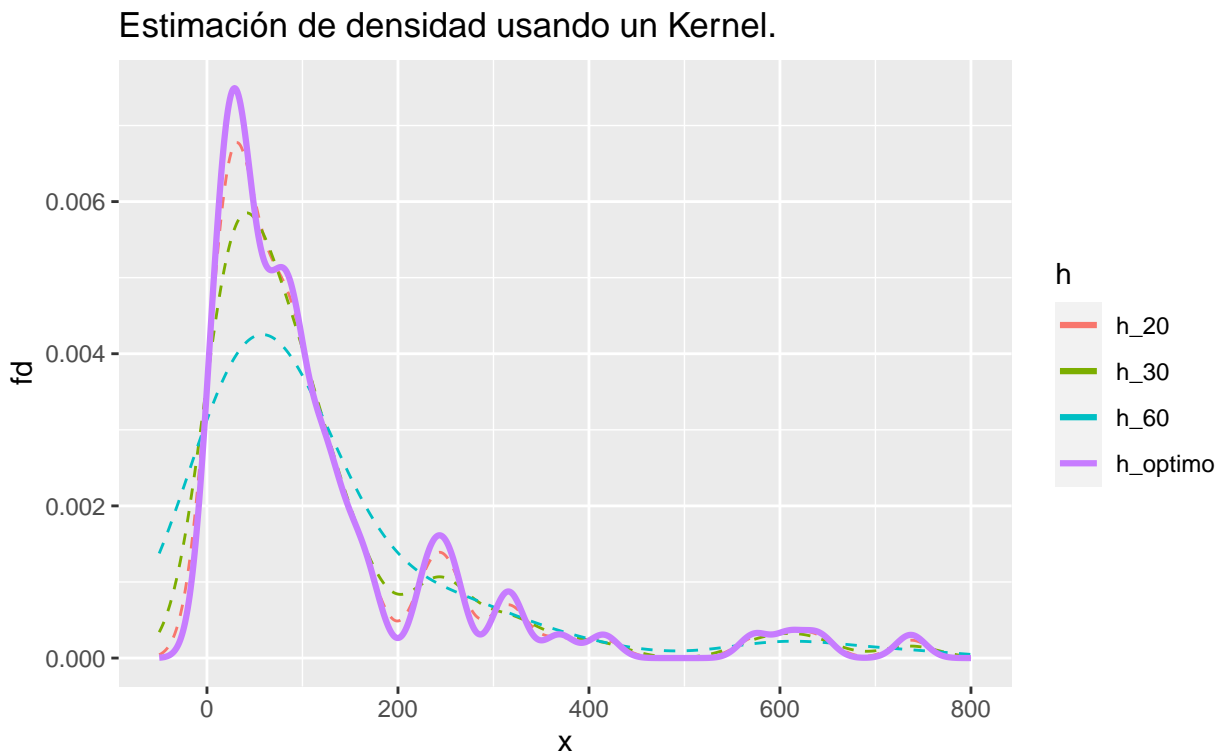


```
## $minimum
## [1] 15.40501
##
## $objective
## [1] -0.004076096
```

Es decir, con  $h = 15.40$  se minimiza el ISE. La gráfica de la densidad estimada con esa  $h$  optima es:

```
x <- seq(-50,800, 1)
fd_optimo <-data.frame(x=x,
                      fd=kde(x, 15.40, "gaussian", tratamiento$X1),
                      h=c(rep("h_optimo", 851)))

ggplot(fd_tratamiento, aes(x=x, y=fd)) +
  geom_line(aes(colour=h), linetype=2) +
  geom_line(data=fd_optimo, aes(x=x,y=fd, colour=h),linetype=1, size=1.1)+
  labs(title = "Estimación de densidad usando un Kernel.")
```



8. En este ejercicio se comprobará que tan buena es la aproximación dada por las reglas empíricas para algunas de las distribuciones estudiadas en la clase. Considere las distribuciones  $Unif(a = -3, b = 3)$ ,  $Normal(0, 1)$ ,  $Exponencial(2)$ ,  $Gamma(\alpha = 2, \beta = 1)$ ,  $Gamma(\alpha = 3, \beta = 1)$ ,  $Beta(\alpha = 2, \beta = 2)$ ,  $Weibull(\alpha = 4, \beta = 1)$  y  $Lognormal(\mu = 3, \sigma = 2)$ .
- a) Para cada una de las distribuciones anteriores, haga una tabla que muestre las probabilidades contenidas en los intervalos  $(\mu - k\sigma, \mu + k\sigma)$ , para  $k = 1, 2, 3$ . Utilice las fórmulas de las medias y varianzas contenidas en las notas para determinar  $\mu$  y  $\sigma$  en cada caso. Puede usar R para determinar las probabilidades pedidas.

## RESPUESTA

Determinemos para cada distribución su media y varianza las cuales se pueden calcular con la table

Distribución de $X$	$\mathbb{E}[X]$	$\text{Var}(X)$
$Unif(a, b)$	$\frac{b+a}{2}$	$\frac{(a-b)^2}{12}$
$Normal(\mu, \sigma)$	$\mu$	$\sigma^2$
$Exponencial(\theta)$	$\theta$	$\theta^2$
$Gamma(\alpha, \beta)$	$\alpha\beta$	$\alpha\beta^2$
$Beta(\alpha, \beta)$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta^2)(\alpha+\beta+1)}$
$Weibull(\alpha, \beta)$	$\alpha^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}\right)$	$\alpha^{-\frac{1}{\beta}}\left(\Gamma\left(\frac{2}{\beta}+1\right)-\Gamma^2\left(\frac{1}{\beta}+1\right)\right)$
$LogNormal(\mu, \sigma)$	$e^{\mu+\frac{\sigma^2}{2}}$	$e^{2\mu+\sigma^2}\left(e^{\sigma^2}-1\right)$

Para este problema sería:

Distribución de $X$	$\mathbb{E}[X]$	$\text{Var}(X)$	$\sigma$
$Unif(-3, 3)$	$\frac{3-3}{2} = 0$	$\frac{(a-b)^2}{12} = \frac{(-3-3)^2}{12} = 3$	1.73
$Normal(0, 1)$	0	1	1
$Exponencial(2)$	2	4	2
$Gamma(2, 1)$	2	2	1.41
$Gamma(3, 1)$	3	3	1.73
$Beta(2, 2)$	$\frac{2}{2+2} = \frac{1}{2}$	$\frac{2(2)}{(2+2^2)(2+2+1)} = \frac{4}{30}$	3
$Weibull(\alpha, \beta)$	$\alpha^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}\right)$	$\alpha^{-\frac{1}{\beta}}\left(\Gamma\left(\frac{2}{\beta}+1\right)-\Gamma^2\left(\frac{1}{\beta}+1\right)\right)$	
$LogNormal(3, 2)$	$e^{3+\frac{2^2}{2}} = 148.41$	$e^{2(3)+4}\left(e^3-1\right)$	1086.544

Calculemos las probabilidades:

```

parametros_distribuciones <- data.frame(distribucion=c("Unif", "Normal", "Exp", "Gamma", "Gamma_2",
resultados_uniforme <- parametros_distribuciones %>% filter(distribucion=="Unif")%>%
  mutate(prob_between_x_s=punif(mu+std, min=-3, max=3)-punif(mu-std, min=-3, max=3),
    prob_between_x_2s=punif(mu+2*std, min=-3, max=3)-punif(mu-2*std, min=-3, max=3),
    prob_between_x_3s=punif(mu+3*std, min=-3, max=3)-punif(mu-3*std, min=-3, max=3))

resultados_normal <- parametros_distribuciones %>% filter(distribucion=="Normal")%>%
  mutate(prob_between_x_s=pnorm(mu+std)-pnorm(mu-std),
    prob_between_x_2s=pnorm(mu+2*std)-pnorm(mu-2*std),

```

```

    prob_between_x_3s=pnorm(mu+3*std)-pnorm(mu-3*std))

resultados_exponencial <- parametros_distribuciones %>% filter(distribucion=="Exp")%>%
  mutate(prob_between_x_s=pexp(mu+std, 2)-pexp(mu-std,2),
         prob_between_x_2s=pexp(mu+2*std, 2)-pexp(mu-2*std, 2),
         prob_between_x_3s=pexp(mu+3*std, 2)-pexp(mu-3*std, 2))

resultados_gamma <- parametros_distribuciones %>% filter(distribucion=="Gamma")%>%
  mutate(prob_between_x_s=pgamma(mu+std, 2, 1)-pgamma(mu-std, 2, 1),
         prob_between_x_2s=pgamma(mu+2*std, 2, 1)-pgamma(mu-2*std, 2, 1),
         prob_between_x_3s=pgamma(mu+3*std, 2, 1)-pgamma(mu-3*std, 2, 1))

resultados_gamma2 <- parametros_distribuciones %>% filter(distribucion=="Gamma")%>%
  mutate(prob_between_x_s=pgamma(mu+std, 3, 1)-pgamma(mu-std, 3, 1),
         prob_between_x_2s=pgamma(mu+2*std, 3, 1)-pgamma(mu-2*std, 3, 1),
         prob_between_x_3s=pgamma(mu+3*std, 3, 1)-pgamma(mu-3*std, 3, 1))

resultados_beta <- parametros_distribuciones %>% filter(distribucion=="Beta")%>%
  mutate(prob_between_x_s=pgamma(mu+std, 2, 2)-pgamma(mu-std, 2, 2),
         prob_between_x_2s=pgamma(mu+2*std, 2, 2)-pgamma(mu-2*std, 2, 2),
         prob_between_x_3s=pgamma(mu+3*std, 2, 2)-pgamma(mu-3*std, 2, 2))

resultados_weibull <- parametros_distribuciones %>% filter(distribucion=="Weibull")%>%
  mutate(prob_between_x_s=pgamma(mu+std, 4, 1)-pgamma(mu-std, 3, 1),
         prob_between_x_2s=pgamma(mu+2*std, 4, 1)-pgamma(mu-2*std, 4, 1),
         prob_between_x_3s=pgamma(mu+3*std, 4, 1)-pgamma(mu-3*std, 4, 1))

resultados_lnorm <- parametros_distribuciones %>% filter(distribucion=="LNormal")%>%
  mutate(prob_between_x_s=pgamma(mu+std, 3, 2)-pgamma(mu-std, 3, 2),
         prob_between_x_2s=pgamma(mu+2*std, 3, 2)-pgamma(mu-2*std, 3, 2),
         prob_between_x_3s=pgamma(mu+3*std, 3, 2)-pgamma(mu-3*std, 3, 2))

resultados_teoricos <- rbind(resultados_beta, resultados_exponencial, resultados_gamma, resultados_gamma2,
                             resultados_lnorm, resultados_normal, resultados_uniforme, resultados_weibull)

resultados_teoricos %>% select(distribucion, prob_between_x_s, prob_between_x_2s, prob_between_x_3s)

##  distribucion prob_between_x_s prob_between_x_2s prob_between_x_3s
## 1      Beta      0.461211399      0.68452660      0.8084851
## 2      Exp      0.999664537      0.9999386      0.9999999
## 3      Gamma    0.735668689      0.95305130      0.9857609
## 4      Gamma    0.640046162      0.85934589      0.9475408
## 5      LNormal  1.000000000      1.00000000      1.0000000
## 6      Normal   0.841344746      0.95449974      0.9973002
## 7      Unif     0.576666667      1.00000000      1.0000000
## 8      Weibull  -0.002236385      0.06389083      0.1006770

```

- b) En R, simule  $n = 1000$  muestras de cada una de las distribuciones anteriores y calcule la media muestral  $\bar{x}$  y la varianza muestral  $s^2$  como se mencionó en la clase. En cada caso, calcule la proporción de observaciones que quedan en los intervalos  $(\bar{x} - k, \bar{x} + ks)$ , para  $k = 1, 2, 3$ . Reporte sus hallazgos

en una tabla como la del inciso anterior. ¿Qué tanto se parecen la tabla de este inciso y la del anterior?

## RESPUESTA

Calculemos

```
set.seed(08081997)

muestras_simuladas<- data.frame(x=c(runif(1000, -3, 3), rnorm(1000), rexp(1000, 2),
  rgamma(1000, 2, 1), rgamma(1000, 3, 1), rbeta(1000, 2, 2), rweibull(1000, 4, 1),
  rlnorm(1000,meanlog= 3, sdlog = 2)), distribucion=c(rep("Uniforme", 1000), rep("Normal", 1000),
  rep("Exp", 1000), rep("Gamma_1", 1000), rep("Gamma_2", 1000), rep("Beta", 1000),
  rep("Weib", 1000), rep("LNormal", 1000)))

estadisticos_muestrales <- muestras_simuladas %>% group_by(distribucion) %>%
  summarise(mean_muestral = mean(x),
    std_muestral = sqrt(var(x)))

head(estadisticos_muestrales,8)

## # A tibble: 8 x 3
##   distribucion mean_muestral std_muestral
##   <fct>          <dbl>          <dbl>
## 1 Beta           0.505           0.234
## 2 Exp            0.523           0.505
## 3 Gamma_1        2.04           1.49
## 4 Gamma_2        2.97           1.67
## 5 LNormal       152.           562.
## 6 Normal         0.0111          0.998
## 7 Uniforme      -0.0369           1.74
## 8 Weib           0.907           0.257

resultados <- merge(muestras_simuladas, estadisticos_muestrales)

resultados %>% mutate(
  x_s = ifelse(x>(mean_muestral-std_muestral) & x<(mean_muestral+std_muestral), 1, 0),
  x_2s = ifelse(x>(mean_muestral-2*std_muestral) & x<(mean_muestral+2*std_muestral), 1, 0),
  x_3s = ifelse(x>(mean_muestral-3*std_muestral) & x<(mean_muestral+3*std_muestral), 1, 0)) %>%
  group_by(distribucion) %>% summarise(propor_x_s = sum(x_s)/1000,
    propor_x_2s = sum(x_2s)/1000,
    propor_x_3s = sum(x_3s)/1000)

## # A tibble: 8 x 4
##   distribucion propor_x_s propor_x_2s propor_x_3s
##   <fct>          <dbl>          <dbl>          <dbl>
## 1 Beta           0.619           0.992           1
## 2 Exp            0.829           0.942           0.988
## 3 Gamma_1        0.75           0.955           0.986
## 4 Gamma_2        0.711           0.953           0.988
## 5 LNormal        0.961           0.976           0.987
## 6 Normal         0.672           0.962           0.999
## 7 Uniforme       0.567           1             1
```

```
## 8 Weib          0.697          0.951          0.999
```

```
resultados_teoricos %>% select(distribucion, prob_between_x_s, prob_between_x_2s, prob_between_x_3s)
```

```
##   distribucion prob_between_x_s prob_between_x_2s prob_between_x_3s
## 1      Beta      0.461211399      0.68452660      0.8084851
## 2      Exp      0.999664537      0.99999386      0.9999999
## 3      Gamma    0.735668689      0.95305130      0.9857609
## 4      Gamma    0.640046162      0.85934589      0.9475408
## 5    LNormal    1.000000000      1.00000000      1.0000000
## 6     Normal    0.841344746      0.95449974      0.9973002
## 7      Unif     0.576666667      1.00000000      1.0000000
## 8    Weibull    -0.002236385      0.06389083      0.1006770
```

## Bibliografía

Härdle, W.K., D.R. Brillinger, S.E. Fienberg, J. Gani, J.A. Hartigan, J.C. Kiefer, and K. Krickeberg. 1991. *Smoothing Techniques: With Implementation in S*. Springer Series in Statistics. Springer. [https://books.google.ne/books?id=2dxxO/\\_zyRagC](https://books.google.ne/books?id=2dxxO/_zyRagC).

Weglarczyk, Stanislaw. 2018. “Kernel Density Estimation and Its Application.” *ITM Web of Conferences* 23 (January): 00037. <https://doi.org/10.1051/itmconf/20182300037>.