Aprendizaje Estadístico - (84:04)

Doctorado en Ingeniería Civil

Alejandro Verri Kozlowski

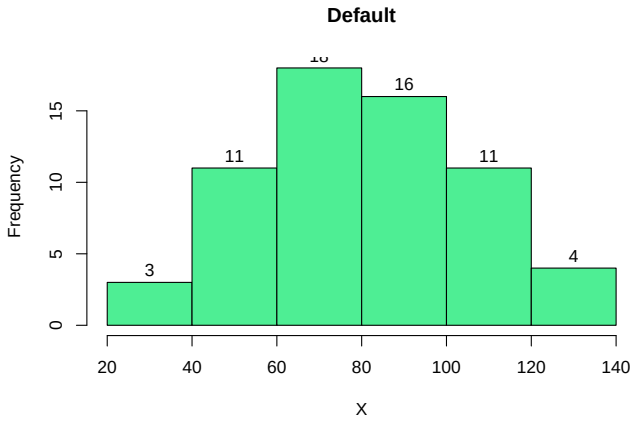
1 Estimación no-paramétrica

1. En el archivo `buffalo.txt` se encuentran los datos que corresponden a la mediciones de cantidad de nieve caída (en pulgadas) en Buffalo en los inviernos de 1910/1911 a 1972/1973.Realizar un histograma para estos datos utilizando los parámetros por defecto.

```
DT <- fread("data/buffalo.txt", sep=" ") |> transpose()
X <- DT$V1 |> sort()
summary(X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   64.50   79.60   80.30   97.65  126.40
```

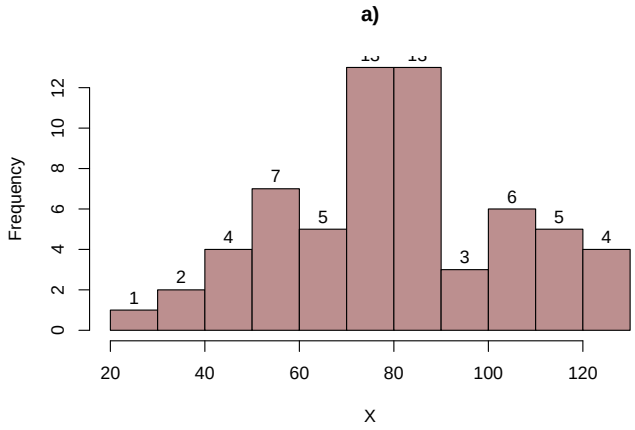
```
hist(X, main="Default",col = "seagreen2",labels = TRUE)
```



Repetir eligiendo como puntos de corte las siguientes secuencias

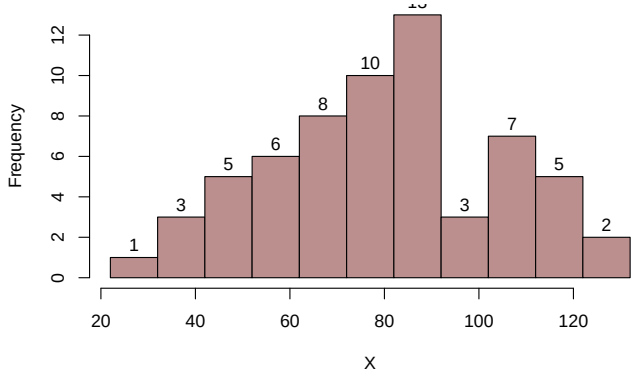
a. De 20 a 130 con un paso de 10

```
BRK <- seq(from=20,to=130,by=10)
hist(X,breaks=BRK, main="a)", col = "rosybrown", labels = TRUE)
```



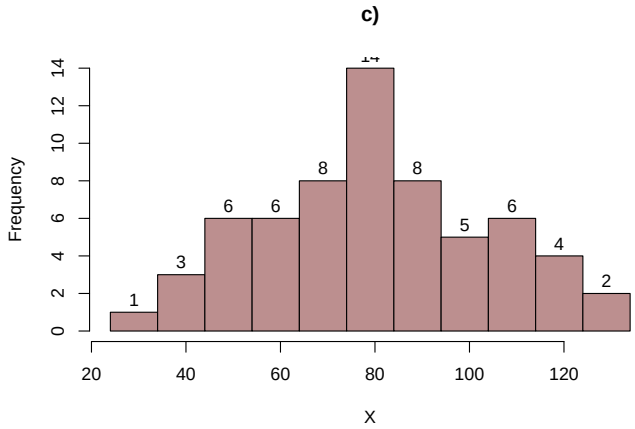
b. De 22 a 132 con un paso de 10

```
BRK <- seq(from=22,to=132,by=10)
hist(X,breaks=BRK,main=NULL, col = "rosybrown", labels = TRUE)
```



c. De 24 a 134 con un paso de 10

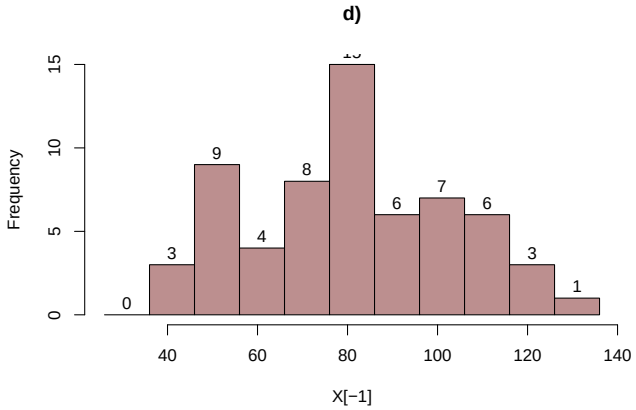
```
BRK <- seq(from=24,to=134,by=10)
hist(X,breaks=BRK, main="c)", col = "rosybrown", labels = TRUE)
```



d. De 26 a 136 con un paso de 10 [1]

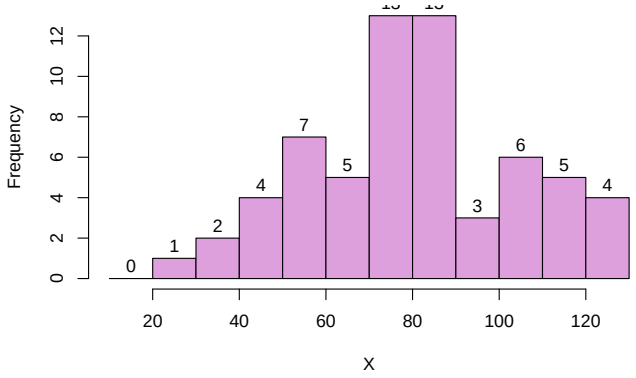
[1] Este histograma no puede visualizarse con las propiedades por defecto de la función `hist()`. El valor mínimo de `X` 25 queda fuera del límite inferior de los puntos de corte. PAra poder visualizar este histograma debe excluirse el primer punto del vector aleatorio X

```
BRK <- seq(from=26,to=136,by=10)
hist(X[-1],breaks=BRK, main="d)", col = "rosybrown", labels = TRUE)
```



2. Realizar un histograma para estas observaciones utilizando puntos de corte `(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130)`. Comparar todos los histogramas obtenidos. ¿Tiene algún efecto la elección del punto inicial para estos datos?

```
BRK <- seq(from=10,to=130,by=10)
hist(X,breaks=BRK,main="", col = "plum", labels = TRUE)
```



Los histogramas con el mismo ancho de contenedor h (binwidth) son diferentes y dependen fuertemente de la elección del origen x_0. Para un mismo h, la elección del punto inicial x_0 cambia completamente la forma (distribución de frecuencias) del histograma.

3. Sea X la cantidad de nieve caída en un invierno en Buffalo. Implementar una función que dados los valores x, h y un conjunto de datos X, permita estimar a $P[X \in (x-h, x+h)] = P[x-h \leq X \leq x+h]$ para cada valor x.

Solución: Si X es una V.A.C, la probabilidad de X esté comprendida en el intervalo abierto $(x-h, x+h)$ puede estimarse mediante

$$P[X \in (x-h, x+h)] \approx n^{-1} \sum_{i=1}^{n} \boldsymbol{I}[X_i \in (x-h, x+h)]$$

donde $\boldsymbol{I}[X_i \in (x-h, x+h)]$ *es la función identificadora, que devuelve* `1` *si* X_i *pertenece al intervalo y* `0` *de otro modo. La función* `Px.h (x,X,h)` *permite estimar la expresión anterior, tanto para valores de* x *escalares, como así también asumiendo* $x = X$ *y calculando las probabilidades (densidades) de todos los puntos del intervalo*

```
Px.h <- function(x,X,h){
  n <- length(X)
  sapply(x, function(t){sum(X>(t-h) &  X<(t+h))/n})}
# Chequear con Jemina los intervalos abiertos
```

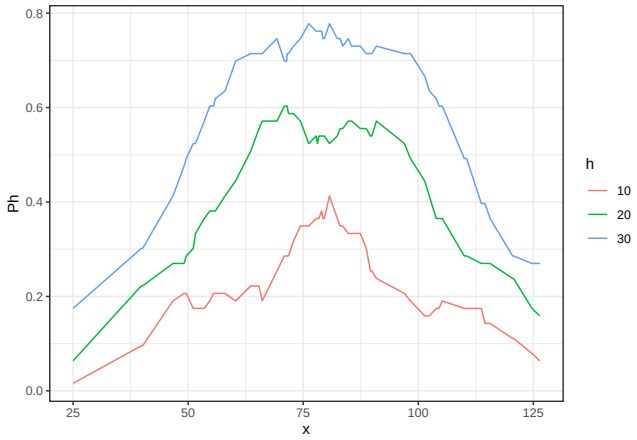
4. Calcular la estimación de la probabilidad definida en el ítem anterior para cada valor x de la muestra usando h 10, 20 y 30.

   La tabla siguiente muestra las probabilidades estimadas mediante la función `Px.h (x,X,h)` *para los 10 primeros valores del vector* `X`

```
TABLE <- data.table(
  "x[i]"=X,
  "Px(h=10)"=Px.h(x=X,X=X,h=10) |> round(digits = 4),
  "Px(h=20)"=Px.h(x=X,X=X,h=20) |> round(digits = 4),
  "Px(h=30)"=Px.h(x=X,X=X,h=30)|> round(digits = 4)
)
head(TABLE,10) |> flextable()
```

x[i]	Px(h=10)	Px(h=20)	Px(h=30)
25.0	0.0159	0.0635	0.1746
39.8	0.0952	0.2222	0.3016
39.9	0.0952	0.2222	0.3016
40.1	0.0952	0.2222	0.3016
46.7	0.1905	0.2698	0.4127
49.1	0.2063	0.2698	0.4762
49.6	0.2063	0.2857	0.4921
51.1	0.1746	0.3016	0.5238
51.6	0.1746	0.3333	0.5238
53.5	0.1746	0.3651	0.5714

```
DATA <- rbindlist(use.names = TRUE,l=list(
  data.table(h=10,x=X,Ph=Px.h(x=X,X=X,h=10)),
  data.table(h=20,x=X,Ph=Px.h(x=X,X=X,h=20)),
  data.table(h=30,x=X,Ph=Px.h(x=X,X=X,h=30)))
)
ggplot(DATA[,.(x,Ph,h=as.factor(h))], aes(x=x, y=Ph, fill=h, color=h)) + geom_line() +  theme_bw()
```



5. Implementar una función `densidad.est.parzen()` que dados un conjunto de datos, una ventana `h` y un punto `x` y devuelva $fh(x)$, el valor de la estimación de la densidad f en el punto x, utilizando el núcleo uniforme (también llamado rectangular).

   Se puede demostrar que una estimación de la densidad $\hat{f}_h(x)$ en un punto x puede obtenerse a partir de la probabilidad $P[X \in (x-h, x+h)]$ según

$$\hat{f}_h(x) \approx \frac{1}{2hn} \sum_{i=1}^{n} \boldsymbol{I}[X_i \in (x-h, x+h)]$$

   Si se efectúa un cambio de variable $t = (x - X_i)/h$ la función indicadora queda definida como: $\boldsymbol{I}[X_i \in (x(t)-h, x(t)+h)] = \boldsymbol{I}[-1 < t < +1]$. Si definimos una función *núcleo* como $K(t) = 1/2\ \boldsymbol{I}[-1 < t < +1]$, el estimador de densidad puede escribirse en términos de dicha función como:_

$$\hat{f}_h(x) \approx \frac{1}{hn} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

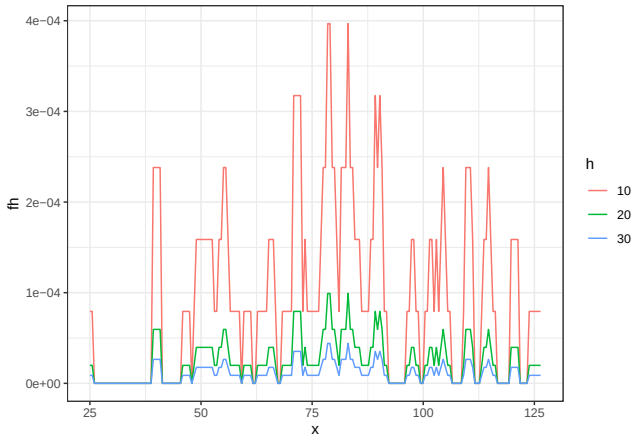
   A partir de esta definición, la siguiente rutina implementa la función del núcleo uniforme Ku(u)y el estimador de densidad ***densidad.est.parzen***

```
Ku <- function(u){  1/2*(u > -1 &  u < +1)}
densidad.est.parzen <- function(x,X,h){
  n <- length(X)
  sapply(x, function(xi){sum(Ku((xi-X))/h)/(n*h)})}
```

6. Con la función `densidad.est.parzen()` implementada, estimar la densidad f en el intervalo `(25,126.4)` (mínimo y máximo de las observaciones) sobre una grilla de 200 puntos equiespaciados para `h = 10, 20, 30`. Graficar las estimaciones obtenidas sobre un mismo gráfico.

```
xgrid <- seq(from=25,to=126.4,length.out=200)

DATA <- rbindlist(use.names = TRUE,l=list(
  data.table(h=10,x=xgrid,fh=densidad.est.parzen(x=xgrid,X=X,h=10)),
  data.table(h=20,x=xgrid,fh=densidad.est.parzen(x=xgrid,X=X,h=20)),
  data.table(h=30,x=xgrid,fh=densidad.est.parzen(x=xgrid,X=X,h=30)))
)
ggplot(DATA[,.(x,fh,h=as.factor(h))]) + geom_line(aes(x=x, y=fh, color=h))+ theme_bw()
```

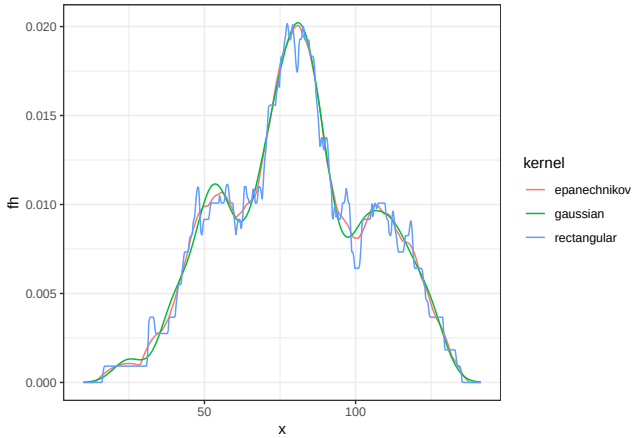


7. La función de R `density()` computa un estimador de la densidad utilizando núcleos a partir de un conjunto de datos y la evalúa en un conjunto de puntos intermedios. Mediante la función de R `density()` estimar la función de densidad $f_h(x)$ a partir de los datos de Buffalo utilizando el núcleo normal, el rectangular y el de Epanechnikov con ventana `h = 5`. Realizar un gráfico superponiendo las tres estimaciones de f y el histograma.

```
G5 <- density(X,kernel = "gaussian",bw=5)
R5 <- density(X,kernel = "rectangular",bw=5)
E5 <- density(X,kernel = "epanechnikov",bw=5)

DATA <- rbindlist(use.names = TRUE,l=list(
  data.table(x=G5$x,fh=G5$y,kernel="gaussian",bw=G5$bw),
  data.table(x=R5$x,fh=R5$y,kernel="rectangular",bw=R5$bw),
 data.table(x=E5$x,fh=E5$y,kernel="epanechnikov",bw=E5$bw)
))
# ggplot(DATA, aes(color=kernel))  + geom_line(aes(x=x, y=fh))+ geom_histogram(aes(x=x),binwidth = 5) + theme_bw()

ggplot(DATA,aes(color=kernel)) + geom_line(aes(x=x,y=fh)) + theme_bw()
```



8. Repetir el ítem anterior para `h = 1,5,10,20,50`. Estudiar el comportamiento de las diferentes estimaciones.

```
DATA <- data.table()
for(KERNEL in c("gaussian","rectangular","epanechnikov")){
```

```
  for(BW in c(5,10,15,20,30,50)){
    D <-  density(X,kernel = KERNEL,bw=BW)
    DT <- data.table(x=D$x,fh=D$y,kernel = KERNEL,bw=as.factor(BW))
    DATA <- rbind(DATA,DT)
  }
}
```