

**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE MATEMÁTICA**



Licenciatura en Estadística

Control Estadístico del Paquete R

**”UNIDAD DOS”
PRÁCTICA 08 - ANÁLISIS ESTADÍSTICO DE DATOS
UNIVARIADOS CONTINUOS EN R..**

**Alumna:
Martha Yoana Medina Sánchez**

**Fecha de elaboración
Santa Ana - 27 de noviembre de 2015**

Práctica 08:Análisis estadístico de datos univariados continuos en R.**ANÁLISIS ESTADÍSTICO DE LOS DATOS.**

1. Visualiza el directorio por defecto y activa su directorio de trabajo

```
getwd()

## [1] "C:/Users/User/Documents/PRACTICAS_YOANA_MEDINA/Yoana/PRACTICAS DE R"

setwd("C:/Users/User/Documents/PRACTICAS_YOANA_MEDINA/Yoana/PRACTICAS DE R")
```

2. Crea un nuevo Script y llámale "Script08-DatosContinuos"
3. Crea el vector que contendrá los datos.

```
Notas <- c(4.47, 4.47); Notas

## [1] 4.47 4.47

data.entry(Notas)
Notas

## [1] 4.47 4.47

length(Notas)

## [1] 2
```

4. Guarda el vector de datos en un archivo.

```
write(Notas, "Notas.txt")
```

5. Limpia el área de trabajo (Workspace)

```
ls()

## [1] "Notas"

rm(list=ls(all=TRUE))
ls()

## character(0)
```

6. Lee o recupera el vector de datos desde el archivo de texto.

```
X <- scan("Notas.txt", what = double(0), na.strings = "NA", flush=FALSE)
ls()

## [1] "X"

# Si el vector contiene valores reales se ocupa:

what = double(0)
```

7. Crea la tabla de frecuencias.

```
# Define el numero k de los intervalos o clases.

# Usa el Metodo de Herbert A. Sturges para determinar dicho numero.

n <- length(X); n

## [1] 2

k <- 1+3.322*logb(n, 10); k

## [1] 2.000022

k <- round(k); k

## [1] 2
```

```
# Calcula el ancho o amplitud a de cada intervalo a=rango/k

rango <- max(X)-min(X); rango

## [1] 0

a=rango/k; a

## [1] 0

a <- round(a, 3); a

## [1] 0
```

```
# Define los limites y puntos mediosde cada uno de los k intervalos

limites <- c(from = min(X)-0.01/2, to = max(X)+0.01/2, by=a); limites

## from to by
## 4.465 4.475 0.000

options(digits=4)
```

```
ci <- cbind(1:k); ci

##      [,1]
## [1,]    1
## [2,]    2

for(i in 2:length(limites)) ci[i-1, 1] <- (limites[i]+ limites[i-1])/2

ci

##      [,1]
## [1,] 4.470
## [2,] 2.237
```

```
# Encuentra las frecuencias absolutas fi para cada intervalo.

options(digits=2)
fi <- cbind(table(cut(X, breaks = limites, labels=NULL, include.lowest=FALSE,
right=FALSE, dig.lab=4)))); fi

##      [,1]
## [0,4.465)    0
## [4.465,4.475)  2
```

```
# breakses un vector o secuencia de cortes 1:6, o el numero de clases.
```

```
# labelsindica que no hay nombres para los intervalos o clases, por defecto
# las etiquetas tienen la notacion (a, b].
```

```
# include.lowestindica que si un  $X[i]$  es igual al corte inferior (0 superior,
# para right=FALSE) el valor debe ser incluido.
```

```
# rightindica que si el intervalo debe ser cerrado a la derecha y abierto a la
# izquierda, o viceversa.
```

```
# dig.labes un entero el cual es usado cuando las etiquetas no son dadas,
# determina el numero de digitos usado en el formato de numeros de cortes.
```

```
# Encuentra las frecuencias relativas o proporciones fri.
```

```
options(digits=4)
fri <- fi/n; fri
```

```
##                [,1]
## [0,4.465)        0
## [4.465,4.475)    1
```

```
# Encuentra las frecuencias acumuladas ascendentes Fi
```

```
options(digits=2)
Fi <- cumsum(fi); Fi
```

```
## [1] 0 2
```

```
# Encuentra las frecuencias relativas acumuladas Fri
```

```
options(digits=4)
Fri <- Fi/n; Fri
```

```
## [1] 0 1
```

```
# Completa la tabla de frecuencias.
```

```
tablaFrec <- data.frame(ci=ci, fi=fi, fri=fri, Fi=Fi, Fri=Fri); tablaFrec
```

```
##                ci fi fri Fi Fri
## [0,4.465)      4.470  0   0  0
## [4.465,4.475)  2.237  2   2  1
```

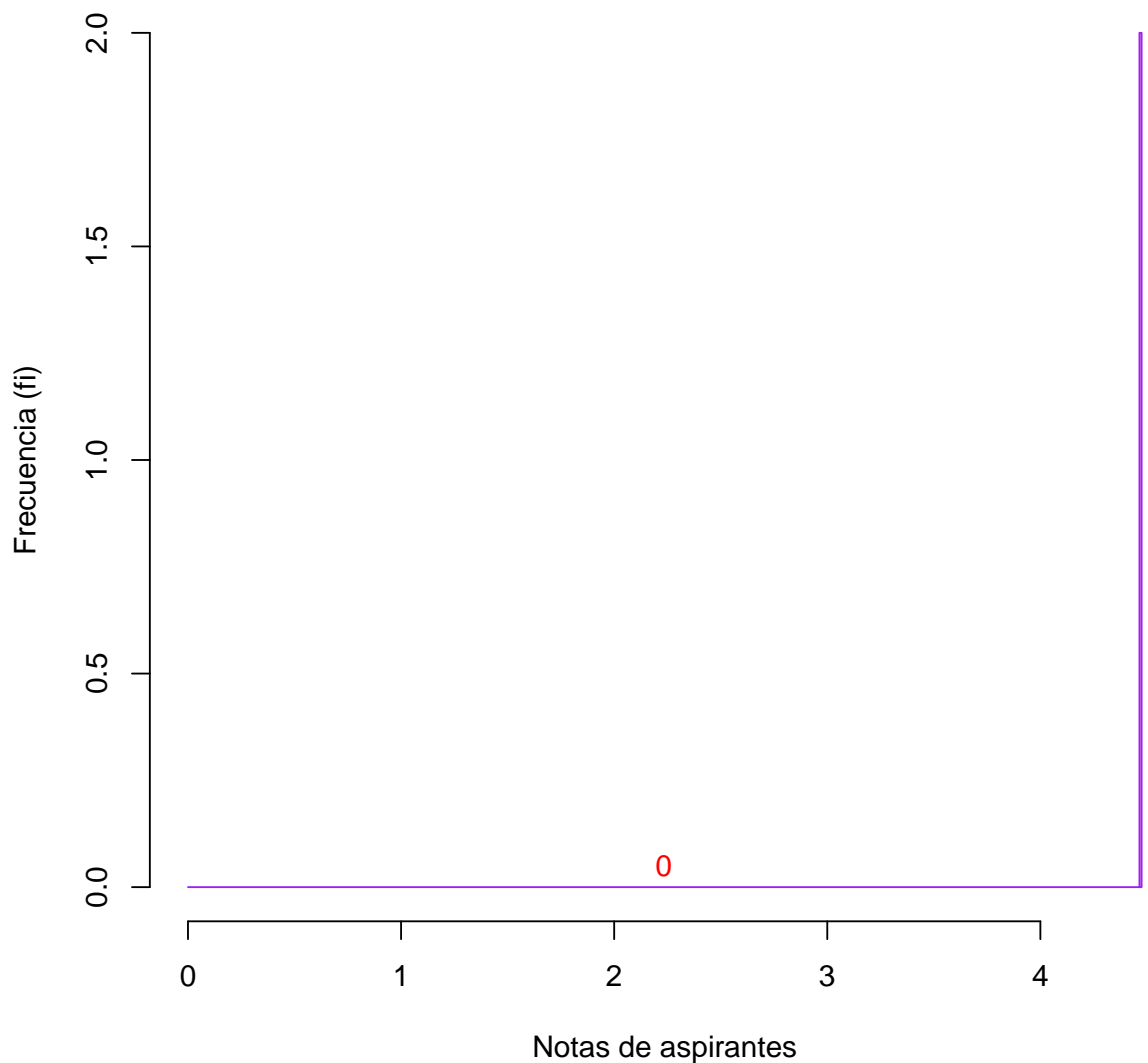
```
# Nuevamente puede usar el comando xtable para importar a codigo LATEX.
```

8. Crea el histograma de frecuencias

```
h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a),
          freq = TRUE, probability = FALSE, include.lowest = FALSE,
          right = TRUE, main = "Histograma de frecuencias",
          col="lightyellow", lty=1, border="purple", xlab="Notas de aspirantes",
          ylab="Frecuencia (fi)", axes=TRUE, labels=FALSE)

## Warning in plot.histogram(r, freq = freq1, col = col, border = border, angle
= angle, : the AREAS in the plot are wrong -- rather use 'freq = FALSE'

text(h$mids, h$density, h$counts, adj=c(0.5, -0.5), col="red")
rug(jitter(X))
```

Histograma de frecuencias

```

# adiciona marcas de los datos
# h es un objeto del tipo lista que contiene atributos del histograma

is.list(h); h

## [1] TRUE
## $breaks
##      by      by from from      to
## 0.000 0.000 4.465 4.465 4.475
##
## $counts
## [1] 0 0 0 2
##
## $density
##      by from from      to
##   NaN      0   NaN   100
##
## $mids
##      by from from      to
## 0.000 2.232 4.465 4.470
##
## $xname
## [1] "X"
##
## $equidist
## [1] FALSE
##
## attr("class")
## [1] "histogram"

```

9. Aproxima al histograma la función de densidad normal

```

h <- hist(X, breaks=c(limite[1]-a, limite, limite[k+1]+a),
          freq = FALSE, probability = TRUE, include.lowest = FALSE,
          right = TRUE, main="Aproximacion a una Normal\n",
          col="lightyellow",lty=1,border="purple",
          xlab="Notas de aspirantes\n", ylab="Frecuencia relativa (fri)",
          axes=TRUE, labels=FALSE)

## Error in plot.window(xlim, ylim, "", ...): se necesitan valores finitos de
## 'ylim'

```

```
text(h$mids, h$density, h$counts, adj=c(0.5, 0.2), col="red")
rug(jitter(X)) # adiciona marcas de los datos

## Warning in rug(jitter(X)): some values will be clipped

curve(dnorm(x, mean=mean(X), sd=sd(X)), col = 2, lty = 2, lwd = 2, add = TRUE)
```

10. Crea el polígono de frecuencias

```
h<- hist(X, breaks=c(limites[1]+a, limites, limites[k+1]+a), freq =TRUE,
           probability=FALSE , include.lowest=FALSE , right=TRUE,
main = "Polígono de frecuencias",col="lightyellow",
```



```
lty=1, border = "purple" , xlab=" Notas de aspirantes" , ylab="Frecuencia (fi)",
axes=TRUE, labels=FALSE)

## Warning in plot.histogram(r, freq = freq1, col = col, border = border, angle
= angle, : the AREAS in the plot are wrong -- rather use 'freq = FALSE'

text(h$mids, h$density, h$counts, adj=c(0.5, -0.5), col="red")
rug(jitter(X)) # adiciona marcas de los datos
vCi <- c(h$mids[1]-a, h$mids, h$mids[k+1]+a); vCi

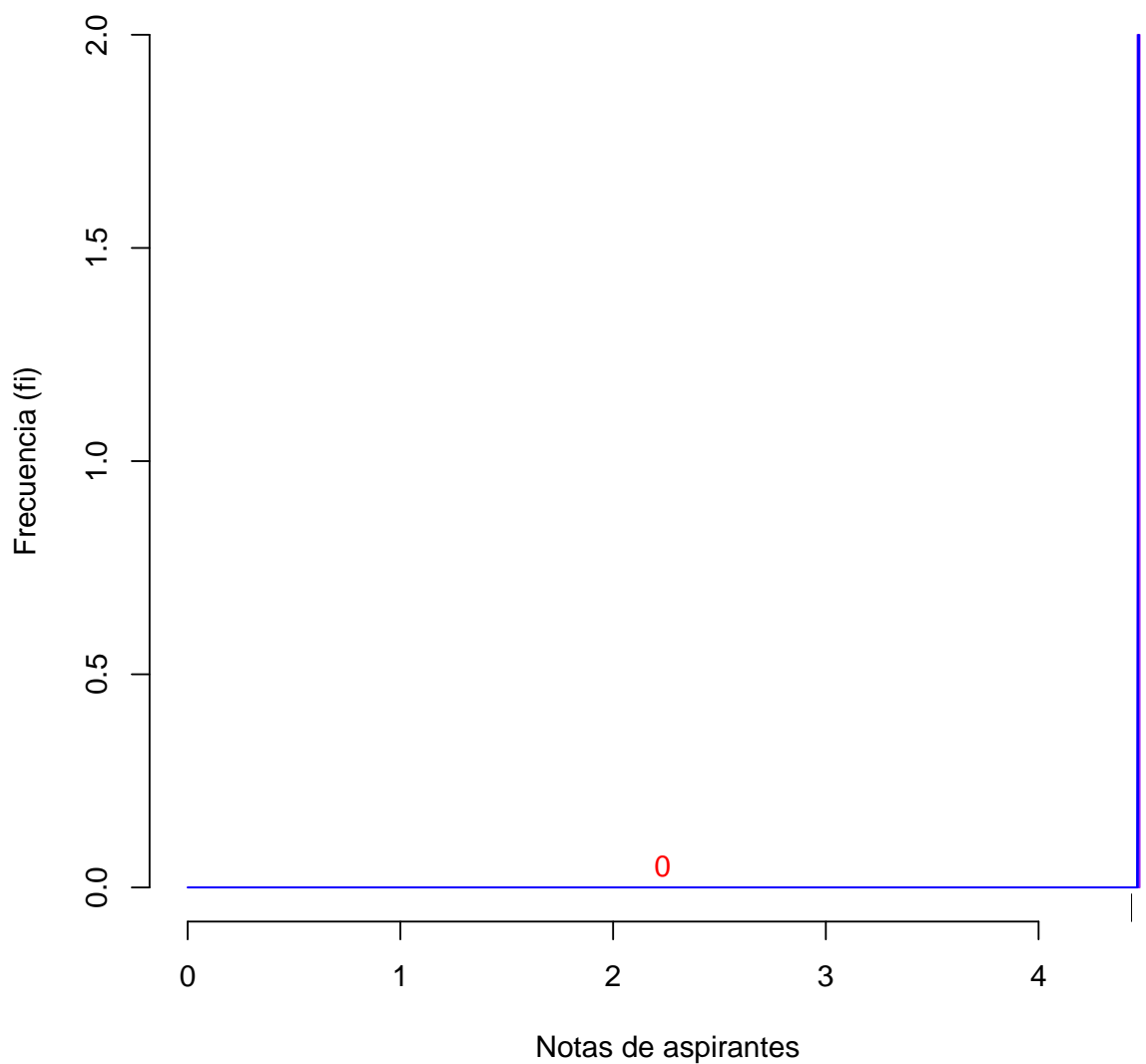
##      by      by from from      to from
## 0.000 0.000 2.232 4.465 4.470 4.465

vfi <- c(0, h$counts, 0); vfi

## [1] 0 0 0 0 2 0

lines(vCi, vfi, col="blue", type="l")
```

Polígono de frecuencias

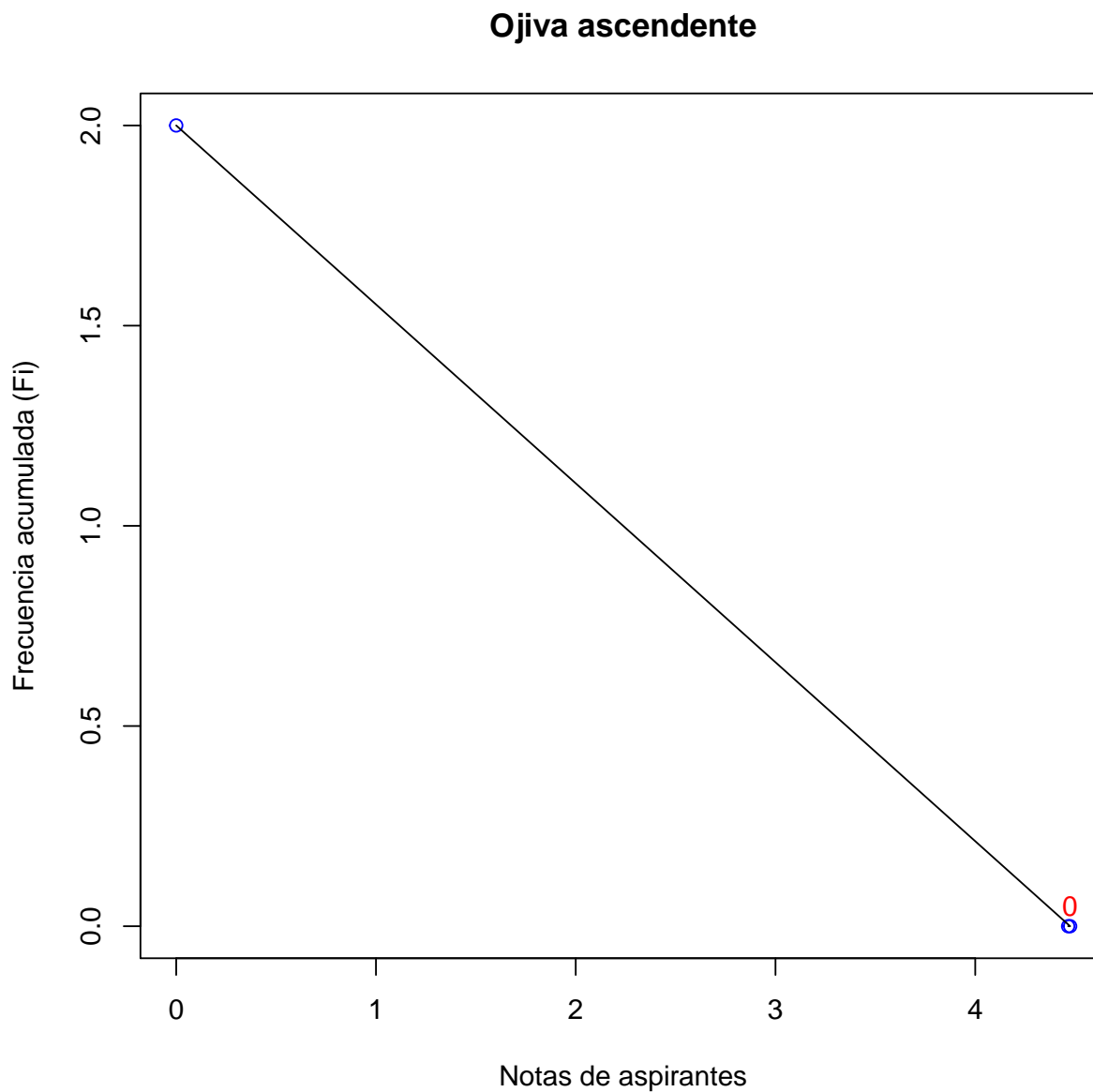


11. Crea la Ojiva ascendente o polígono frecuencias acumuladas ascendentes

```
Fia <- c(0, Fi); Fia

## [1] 0 0 2

plot(limites, Fia, type = "p", pch=1, col = "blue", main="Ojiva ascendente",
xlab="Notas de aspirantes", ylab="Frecuencia acumulada (Fi)")
text(limites, h$density, Fia, adj=c(0.5, -0.5), col="red")
lines(limites, Fia, col="black", type="l")
```



12. Calcula los principales estadístico descriptivos de la variable

```
# Calcula la moda, ya que el R no proporciona una funcion para eso.

options(digits=4)
for(i in 1:k) if (fi[i] == max(fi)) break()
if(i > 1) moda <- limites[i]+((fi[i]-fi[i-1])/((fi[i]-fi[i-1])+(fi[i]-fi[i+1]))))*a

moda

## to
## NA
```