

# GuÃas prÃcticas

Erika MartÃnez

November 27, 2015

UNIDAD 2: Práctica 08 ANÁLISIS ESTADÍSTICO DE LOS DATOS. Crea el vector que contendrá los datos.

```
Notas <- c(4.47, 5.43); Notas  
## [1] 4.47 5.43  
data.entry(Notas)  
Notas  
## [1] 4.47 5.43  
length(Notas)  
## [1] 2
```

Guarda el vector de datos en un archivo.

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

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] "Notas" "X"
```

Crea la tabla de frecuencias. Usa el Método de Herbert A. Sturges para determinar dicho número.

```
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  $a=\text{rango}/k$

```
rango <- max(X)-min(X); rango  
## [1] 0.96  
a=rango/k; a  
## [1] 0.48  
a <- round(a, 3); a  
## [1] 0.48
```

Define los límites y puntos medios de cada uno de los k intervalos

```
limites <- seq(from=min(X)-0.01/2,to=max(X)+0.01/2, by=a); limites
## [1] 4.465 4.945 5.425

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,] 4.705
## [2,] 5.185
```

Encuentra las frecuencias absolutas  $f_i$  para cada intervalo.

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

##      [,1]
## [4.465,4.945)    1
## [4.945,5.425)    0
```

Encuentra las frecuencias relativas o proporciones  $f_{ri}$ .

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

##      [,1]
## [4.465,4.945)  0.5
## [4.945,5.425)  0.0
```

Encuentra las frecuencias acumuladas ascendentes  $F_i$

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

## [1] 1 1
```

Encuentra las frecuencias relativas acumuladas  $F_{ri}$

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

## [1] 0.5 0.5
```

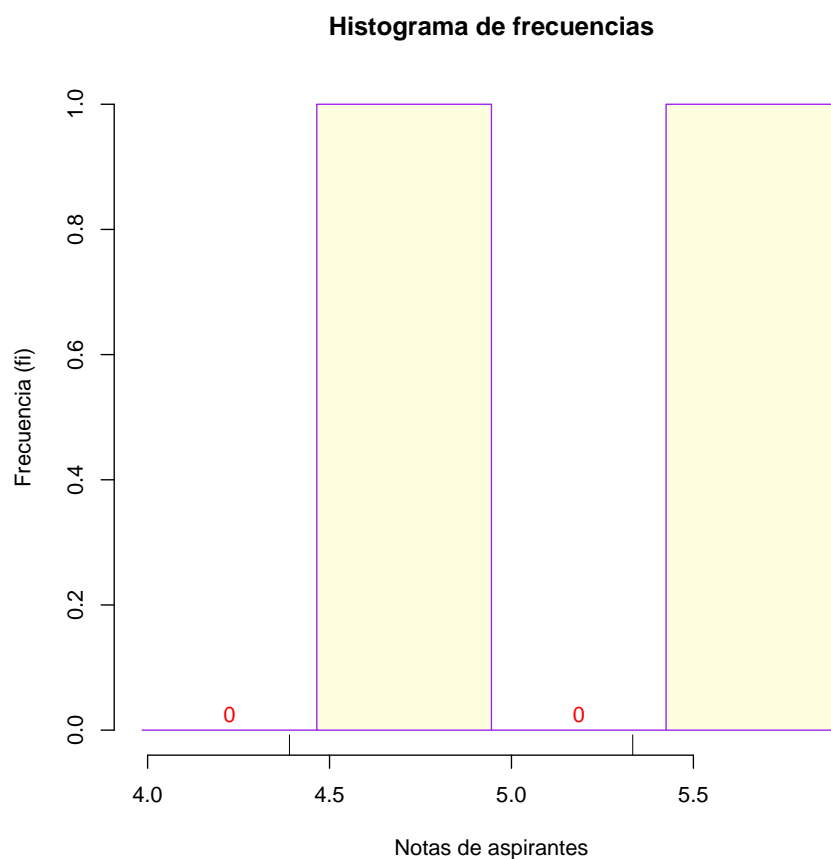
Completa la tabla de frecuencias.

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

##           ci fi fri Fi Fri
## [4.465,4.945) 4.705  1 0.5  1 0.5
## [4.945,5.425) 5.185  0 0.0  1 0.5
```

Crea el histograma de frecuencias

```
h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a), freq = TRUE, probability = F,
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)
text(h$mids, h$density, h$counts, adj=c(0.5, -0.5), col="red")
rug(jitter(Notas))
```



```
is.list(h); h

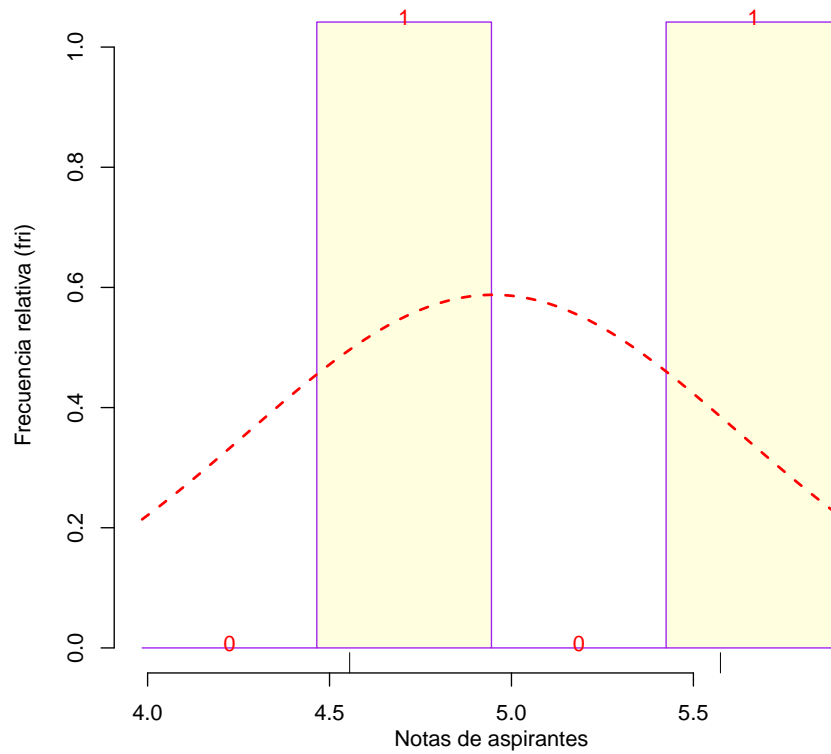
## [1] TRUE
```

```
## $breaks
## [1] 3.985 4.465 4.945 5.425 5.905
##
## $counts
## [1] 0 1 0 1
##
## $density
## [1] 0.000 1.042 0.000 1.042
##
## $mids
## [1] 4.225 4.705 5.185 5.665
##
## $xname
## [1] "X"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

Aproxima al histograma la funci?n de densidad normal

```
h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a), freq = FALSE,
probability = TRUE, include.lowest = FALSE, right = TRUE,
main="Aproximaci?n a una Normal\n", col="lightyellow",lty=1,border="purple",
xlab="Notas de aspirantes\n", ylab="Frecuencia relativa (fri)",
axes=TRUE, labels=FALSE)
text(h$mids, h$density, h$counts, adj=c(0.5, 0.2), col="red")
rug(jitter(X)) # adiciona marcas de los datos
curve(dnorm(x, mean=mean(X), sd=sd(X)), col = 2, lty = 2,lwd = 2, add = TRUE)
```

### Aproximaci?n a una Normal



Crea el pol?gono de frecuencias

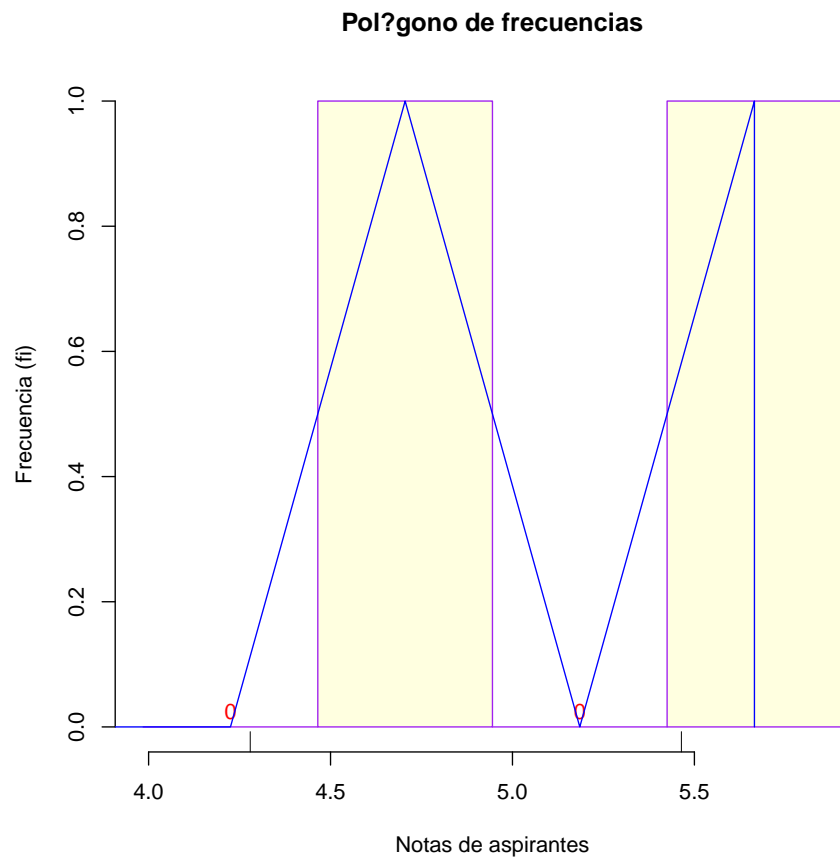
```
h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a), freq = TRUE,
probability=FALSE, include.lowest=FALSE, right=TRUE,
main = "Pol?gono de frecuencias", col="lightyellow", lty=1, border="purple", xlab="
Notas de aspirantes", ylab="Frecuencia (fi)", axes=TRUE, labels=FALSE)
text(h$mids, h$density,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

## [1] 3.745 4.225 4.705 5.185 5.665 5.665

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

## [1] 0 0 1 0 1 0

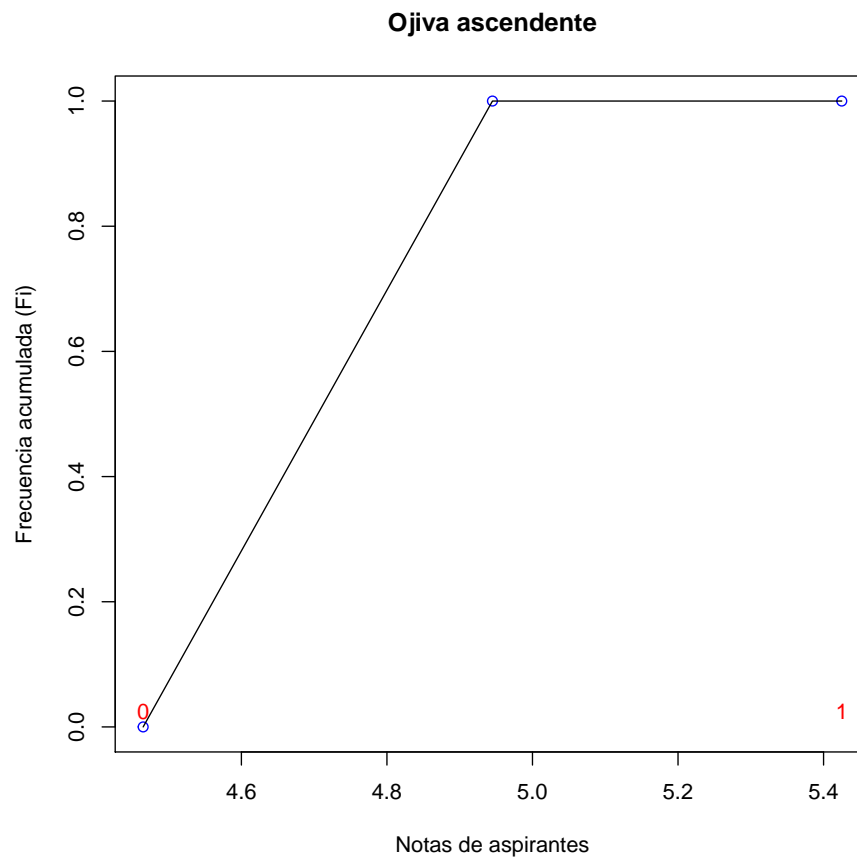
lines(vCi, vfi, col="blue", type="l")
```



Crea la Ojiva ascendente o polígono de frecuencias acumuladas ascendentes

```
Fia <- c(0, Fi); Fia
## [1] 0 1 1

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")
```



Calcula los principales estadísticos descriptivos de la variable. Calcula la moda, ya que el R no proporciona una función 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 else {
moda <- limites[i]+(fi[i]/(fi[i]+(fi[i]-fi[i+1])))*a
moda }

## [1] 4.705
```

Calcula los cuartiles: Q1, Q2, Q3

```
Q <- 1:3
for(v in 1:3) for(i in 1:k) if (Fi[i] > (v*25*n)/100)
{
  Q[v] <- limites[i]+(((25*v*n/100)-Fi[i-1])/fi[i])*a
  break
}

## Error in Q[v] <- limites[i] + (((25 * v * n/100) - Fi[i - 1])/fi[i])
```



```
* : replacement has length zero
Q
```

Calcula los principales estadísticos.

```
estadisticos <- rbind(media=sum(tabEstad$cifi)/n, moda=moda, Q1=Q[1], Q2=Q[2], Q3=Q[3],
rango=max(X)-min(X), varianza=sum(tabEstad$ciMedia2fi)/n,
Desviacion=sqrt(sum(tabEstad$ciMedia2fi)/n),
CoeficienteVariacion=sqrt(sum(tabEstad$ciMedia2fi)/n)/(sum(tabEstad$cifi)/n),
CAfisher=(sum(tabEstad$ciMedia3fi)/n)/sqrt(sum(tabEstad$ciMedia2fi)/n)^3,
CoeficienteCurtosis=((sum(tabEstad$ciMedia4fi)/n)/sqrt(sum(tabEstad$ciMedia2fi)/n)^4)-3)

## Error in rbind(media = sum(tabEstad$cifi)/n, moda = moda, Q1 = Q[1],
Q2 = Q[2], : objeto 'tabEstad' no encontrado

estadisticos

## Error in eval(expr, envir, enclos): objeto 'estadisticos' no encontrado
```

Otros gráficos: Gráfico de cajas

```
boxplot(2, main="Gráfico de caja", xlab="Notas", notch=FALSE,
data=parent.frame(), plot=TRUE, border="red", col="yellow",horizontal=TRUE)
```

**Gráfico de caja**

