

UNIVERSIDAD DE EL SALVADOR

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE



**Universidad de El Salvador**

*Hacia la libertad por la cultura*

PRACTICA 8 DE R

DOCENTE:

LIC. JAIME ISAAC PEÑA

PRESENTADO POR:

CRISTIAN ALBERTO ZALDAÑA ALVARADO

## Índice

1. ANÁLISIS ESTADÍSTICO DE LOS DATOS.	3
---------------------------------------	---

## 1. ANÁLISIS ESTADÍSTICO DE LOS DATOS.

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

```
> getwd()
```

```
[1] "C:/Users/DELL/Documents/Github/PRACTICAS-R-2022"
```

2. Crea el vector que contendrá los datos.

```
> Notas <- c(4.47, 4.47, 3.48, 5.0, 3.42, 3.78, 3.1, 3.57, 4.2, 4.5,  
+ 3.6, 3.75, 4.5, 2.85, 3.7, 4.2, 3.2, 4.05, 4.9, 5.1,  
+ 5.3, 4.16, 4.56, 3.54, 3.5, 5.2, 4.71, 3.7, 4.78, 4.14,  
+ 4.14, 4.8, 4.1, 3.83, 3.6, 2.98, 4.32, 5.1, 4.3, 3.9,  
+ 3.96, 3.54, 4.8, 4.3, 3.39, 4.47, 3.19, 3.75, 3.1, 4.7,  
+ 3.69, 3.3, 2.85, 5.25, 4.68, 4.04, 4.44, 5.43, 3.04, 2.95)  
> Notas
```

```
[1] 4.47 4.47 3.48 5.00 3.42 3.78 3.10 3.57 4.20 4.50 3.60 3.75 4.50 2.85 3.70  
[16] 4.20 3.20 4.05 4.90 5.10 5.30 4.16 4.56 3.54 3.50 5.20 4.71 3.70 4.78 4.14  
[31] 4.14 4.80 4.10 3.83 3.60 2.98 4.32 5.10 4.30 3.90 3.96 3.54 4.80 4.30 3.39  
[46] 4.47 3.19 3.75 3.10 4.70 3.69 3.30 2.85 5.25 4.68 4.04 4.44 5.43 3.04 2.95
```

```
> length(Notas)
```

```
[1] 60
```

3. Guarda el vector de datos en un archivo.

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

4. 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"
```

Si el vector contiene valores reales se ocupa: `what = double(0)`

5. Crea la tabla de frecuencias.

Define el número  $k$  de los intervalos o clases.

Usa el Método de Herbert A. Sturges para determinar dicho número.

```
> n <- length(X)  
> n
```

```
[1] 60
```

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

```
[1] 6.907018
```

```
> k <- round(k)  
> k
```

```
[1] 7
```

Calcula el ancho o amplitud  $a$  de cada intervalo  $a=\text{rango}/k$

```
> rango <- max(X)-min(X)
> rango
```

```
[1] 2.58
```

```
> a=rango/k
> a
```

```
[1] 0.3685714
```

```
> a <- round(a, 3)
> a
```

```
[1] 0.369
```

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] 2.845 3.214 3.583 3.952 4.321 4.690 5.059 5.428
```

```
> options(digits=4)
> ci <- cbind(1:k); ci
```

```
      [,1]
[1,]     1
[2,]     2
[3,]     3
[4,]     4
[5,]     5
[6,]     6
[7,]     7
```

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

```
      [,1]
[1,] 3.030
[2,] 3.399
[3,] 3.768
[4,] 4.136
[5,] 4.505
[6,] 4.875
[7,] 5.244
```

Encuentra las frecuencias absolutas  $f_i$  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]
[2.845,3.214)    9
[3.214,3.583)    8
[3.583,3.952)   10
[3.952,4.321)   12
[4.321,4.69)     8
[4.69,5.059)    7
[5.059,5.428)    5
```

**breaks** es un vector o secuencia de cortes 1:6, o el número de clases.

**labels** indica que no hay nombres para los intervalos o clases, por defecto las etiquetas tienen la notación (a, b].

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

**right** indica que sí el intervalo debe ser cerrado a la derecha y abierto a la izquierda, o viceversa.

**dig.lab** es un entero el cual es usado cuando las etiquetas no son dadas, determina el número de dígitos usado en el formato de números de cortes.

Encuentra las frecuencias relativas o proporciones *fri*.

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

```
          [,1]
[2.845,3.214) 0.15000
[3.214,3.583) 0.13333
[3.583,3.952) 0.16667
[3.952,4.321) 0.20000
[4.321,4.69)  0.13333
[4.69,5.059) 0.11667
[5.059,5.428) 0.08333
```

Encuentra las frecuencias acumuladas ascendentes *Fi*

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

```
[1]  9 17 27 39 47 54 59
```

Encuentra las frecuencias relativas acumuladas *Fri*

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

```
[1] 0.1500 0.2833 0.4500 0.6500 0.7833 0.9000 0.9833
```

Completa la tabla de frecuencias.

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

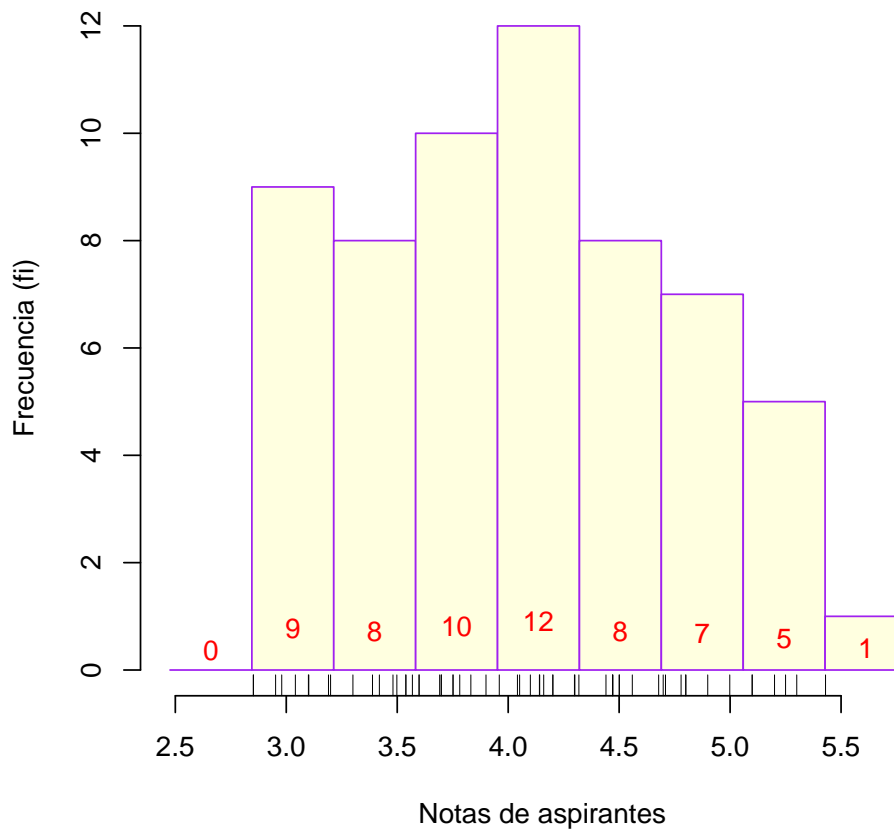
	ci	fi	fri	Fi	Fri
[2.845,3.214)	3.030	9	0.15000	9	0.1500
[3.214,3.583)	3.399	8	0.13333	17	0.2833
[3.583,3.952)	3.768	10	0.16667	27	0.4500
[3.952,4.321)	4.136	12	0.20000	39	0.6500
[4.321,4.69)	4.505	8	0.13333	47	0.7833
[4.69,5.059)	4.875	7	0.11667	54	0.9000
[5.059,5.428)	5.244	5	0.08333	59	0.9833

Nuevamente puede usar el comando xtable para importar a código LATEX.

6. 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)
> text(h$mids, h$density, h$counts, adj=c(0.5, -0.5), col="red")
> rug(jitter(X)) # adiciona marcas de los datos
>
```

### Histograma de frecuencias



h es un objeto del tipo lista que contiene atributos del histograma

```
> is.list(h)
```

```
[1] TRUE
```

```
> h
```

```
$breaks
```

```
[1] 2.476 2.845 3.214 3.583 3.952 4.321 4.690 5.059 5.428 5.797
```

```
$counts
```

```
[1] 0 9 8 10 12 8 7 5 1
```

```
$density
```

```
[1] 0.00000 0.40650 0.36134 0.45167 0.54201 0.36134 0.31617 0.22584 0.04517
```

```
$mids
```

```
[1] 2.660 3.030 3.399 3.768 4.136 4.505 4.875 5.244 5.613
```

```
$xname
```

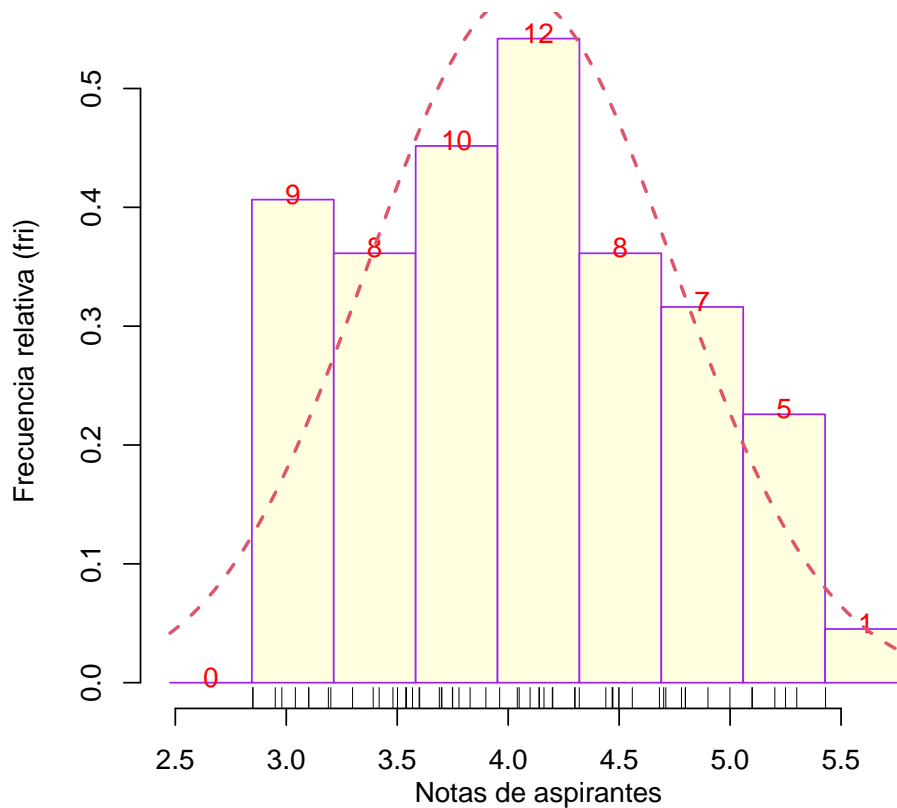
```
[1] "X"
```

```
$equidist  
[1] TRUE  
  
attr(,"class")  
[1] "histogram"
```

7. Aproxima al histograma la función 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="Aproximación 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)
```

### Aproximación a una Normal



8. 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,
```



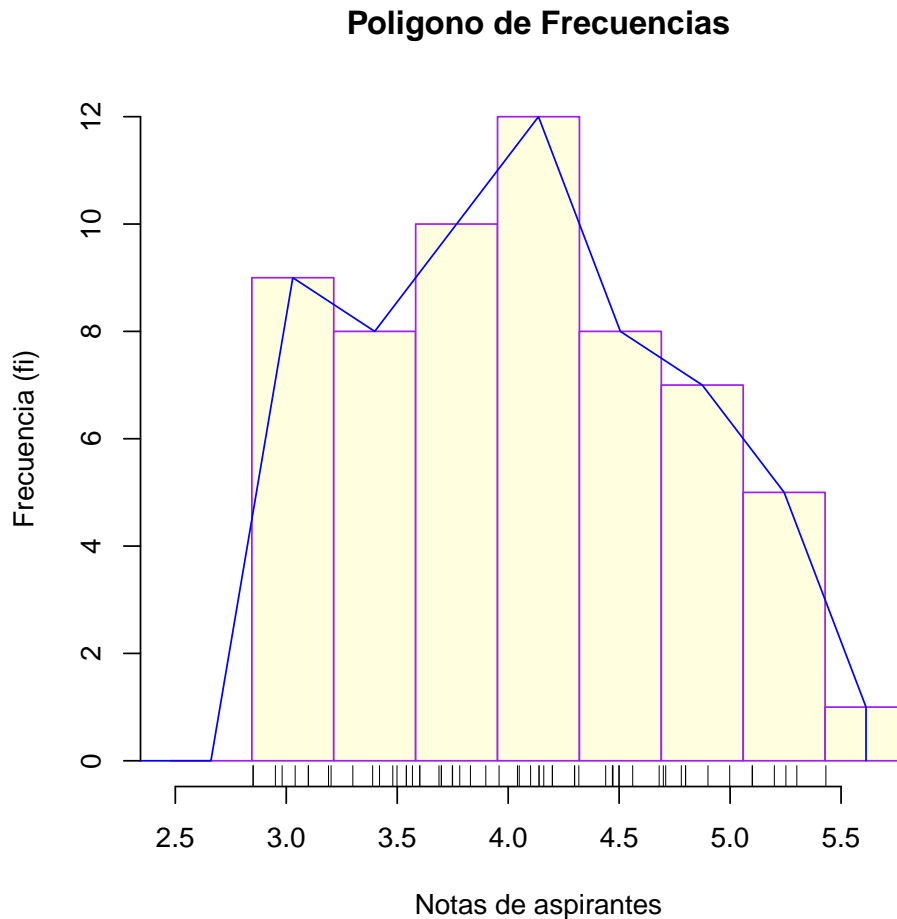
```
+ main="Poligono de Frecuencias",col="lightyellow",lty=1, border="purple",xlab=
+ "Notas de aspirantes",ylab="Frecuencia (fi)",axes=TRUE,labels=FALSE)
> rug(jitter(X)) # adiciona marcas de los datos
> vCi <- c(h$mids[1]-a, h$mids, h$mids[k+1]+a); vCi

[1] 2.292 2.660 3.030 3.399 3.768 4.136 4.505 4.875 5.244 5.613 5.613

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

[1] 0 0 9 8 10 12 8 7 5 1 0

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

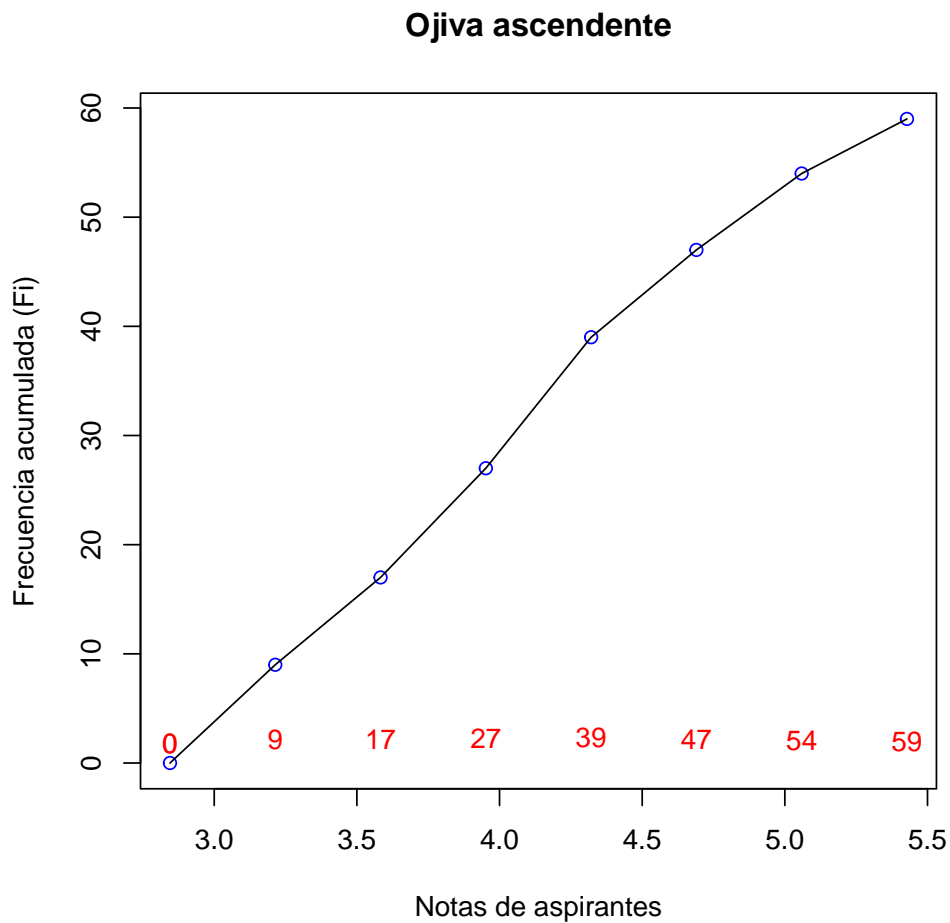


9. Crea la Ojiva ascendente o polígono de frecuencias acumuladas ascendentes

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

[1] 0 9 17 27 39 47 54 59

> 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")
```



10. 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
> moda
```

```
[1] 4.075
```

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
+ }
> Q
```

```
[1] 3.491 4.044 4.598
```

Calcula los principales estadísticos.

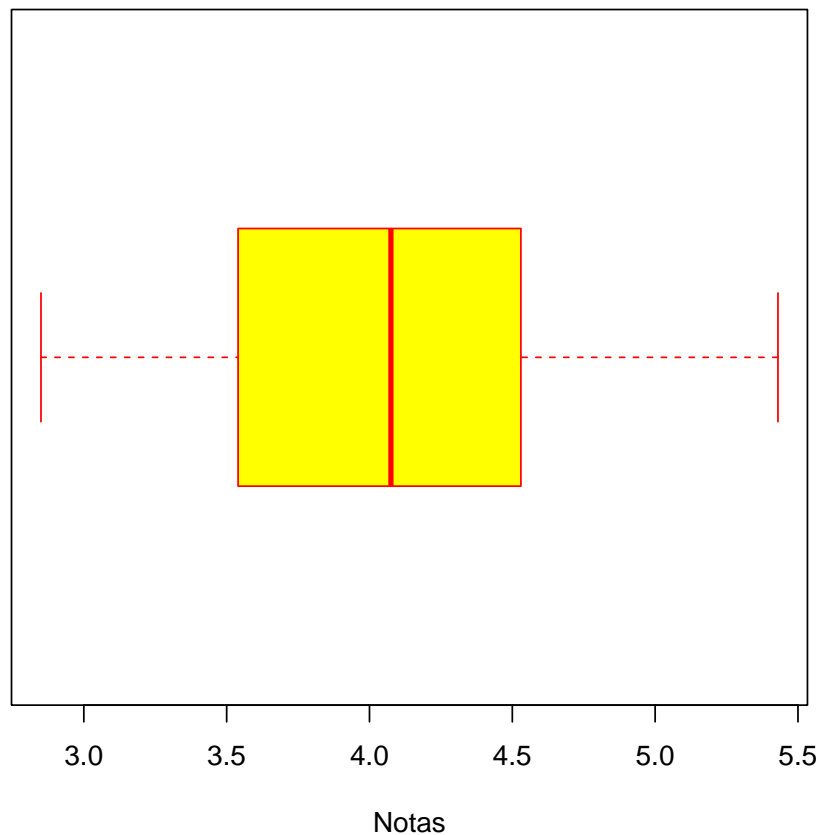
```
> #estadisticos <- rbind(media=sum(tabEstad$cifi)/n, moda=moda, Q1=Q[1], Q2=Q[2], Q3=Q[3],  
> #rango=max(X)-min(X), varianza=sum(tabEstad$ciMedia2fi)/n,  
> #Desviacion=sqrt(sum(tabEstad$ciMedia2fi)/n),  
> #CoeficienteVariacion=sqrt(sum(tabEstad$ciMedia2fi)/n)/(sum(tabEstad$cifi)/n),  
> #CAfisher=(sum(tabEstad$ciMedia3fi)/n)/sqrt(sum(tabEstad$ciMedia2fi)/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

#### 11. Otros gráficos:

Gráfico de cajas

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

**Gráfico de caja**

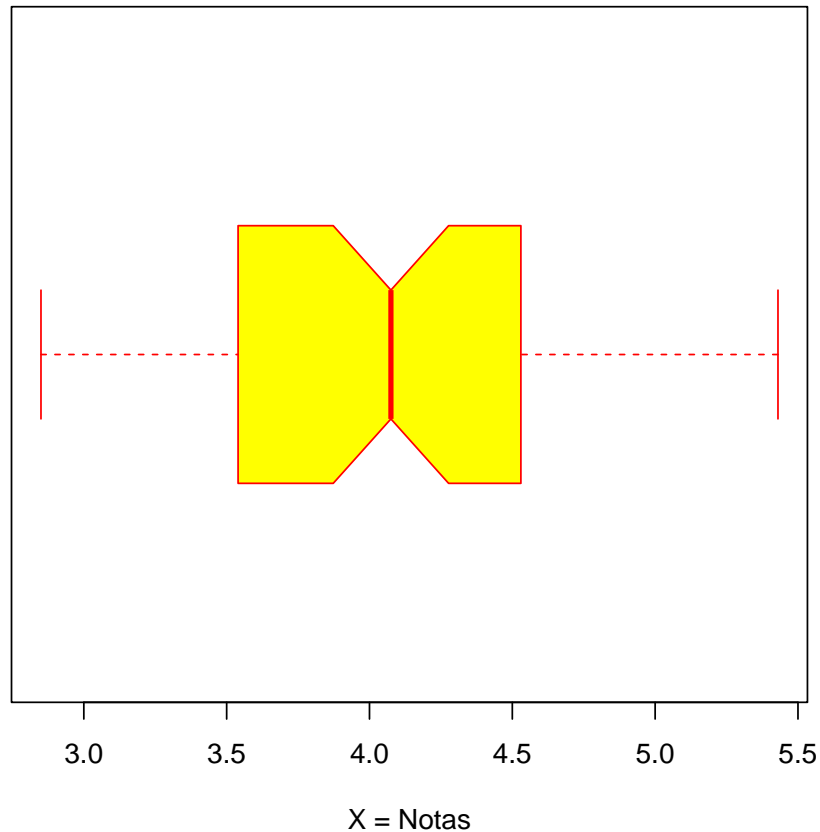


**Observación:** en la función `boxplot()`, si `plot` es `FALSE` se produce un resumen de los valores (los cinco números).

Una variante del boxplot, es el notched boxplot de McGill, Larsen y Tukey, el cual adiciona intervalos de confianza para la mediana, representados con un par de cuñas a los lados de la caja:

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

**Gráfico de caja**



Varios gráficos en una misma ventana

```
> par(mfrow=c(1,2)) # Divide la ventana gráfica en dos partes (1 fila, 2 columnas)  
> hist(X)  
> boxplot(X)
```

