

**UNIVERSIDAD DE EL SALVADOR.**  
**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE.**  
**DEPARTAMENTO DE MATEMATICAS.**



**PRACTICA 07. UNIDAD 2**

**CARRERA:**  
**LICENCIATURA EN ESTADISTICA.**

**ASIGNATURA:**  
**ANALISIS ESTADISTICO CON EL PAQUETE R**

**DOCENTE:**  
**JAIME ISAAC PEÑA**

**PRESENTADO POR:**  
**NELSON DE JESUS MAGAÑA GODINEZ**

**FECHA:**  
**12 AGOSTO DE 2022**

*Ejemplo:* En cierta colonia de San Salvador se selecciona aleatoriamente una muestra de 30 hogares, al medir el número de hijos en cada unidad muestral se obtienen los siguientes datos:

## ANÁLISIS ESTADÍSTICO DE LOS DATOS.

1) Activar el directorio de trabajo

```
getwd()

## [1] "C:/Users/pc 1/Desktop/PAQUETE R/PRACTICAS_S2"

setwd("C:/Users/pc 1/Desktop/PAQUETE R/PRACTICAS_S2")
```

2) Crear un nuevo Script y llamarle "Script07-DatosDiscretos"

3) Crear el vector de datos.

```
Hijos <- c(2,1, 2, 1, 2, 1, 4, 2, 3, 0, 2, 3,
3, 2, 1, 0, 2, 4, 1, 2, 1, 3,
4, 1, 2, 3, 1, 5, 2, 3, 1, 2)
data.entry(Hijos)
Hijos

## [1] 2 1 2 1 2 1 4 2 3 0 2 3 3 2 1 0 2 4 1 2 1 3 4 1 2 3 1 5 2 3 1 2

length(Hijos)

## [1] 32
```

4) Guardar el vector de datos en un archivo de texto.

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

5) Limpiar el área de trabajo (Workspace)

```
ls()

## [1] "Hijos"

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

```

## function (name, pos = -1L, envir = as.environment(pos), all.names = FALSE,
##   pattern, sorted = TRUE)
## {
##   if (!missing(name)) {
##     pos <- tryCatch(name, error = function(e) e)
##     if (inherits(pos, "error")) {
##       name <- substitute(name)
##       if (!is.character(name))
##         name <- deparse(name)
##       warning(gettextf("%s converted to character string",
##         sQuote(name)), domain = NA)
##       pos <- name
##     }
##   }
##   all.names <- .Internal(ls(envir, all.names, sorted))
##   if (!missing(pattern)) {
##     if ((ll <- length(grep("[", pattern, fixed = TRUE))) &&
##       ll != length(grep("]", pattern, fixed = TRUE))) {
##       if (pattern == "[") {
##         pattern <- "\\["
##         warning("replaced regular expression pattern '[' by '\\\\['")
##       }
##       else if (length(grep("[^\\\\\\]\\\\[<-", pattern))) {
##         pattern <- sub("\\\\[<-", "\\\\[<-", pattern)
##         warning("replaced '[<-' by '\\\\[<-' in regular expression pattern")
##       }
##     }
##     grep(pattern, all.names, value = TRUE)
##   }
##   else all.names
## }
## <bytecode: 0x00000196d440c8b0>
## <environment: namespace:base>

```

#### 6) Leer o recuperar el vector de datos o archivo de texto

```

X<- scan("Hijos.txt", what = integer(0), na.strings = "NA",
  flush = FALSE)

ls()

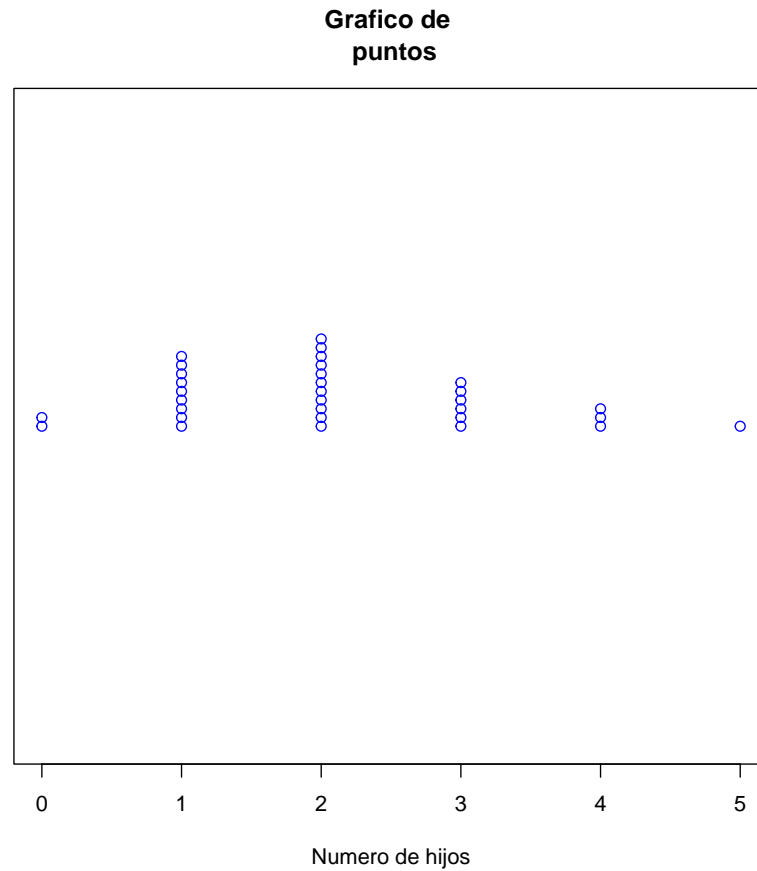
## [1] "X"

# Si el vector contiene caracteres se usa: what = character()
# Si el vector contiene reales se ocupa: what = double(0)

```

7)Elaborar el gráfico de puntos y diagrama de tallo-hojas (stem-and-leaf)

```
# Gráfico de puntos
stripchart(X, method = "stack", vertical = FALSE, col = "blue",
          pch = 1, main="Grafico de \n puntos", xlab = "Numero de hijos")
```



Observación: method puede ser:

”overplot” (los puntos coincidentes son superpuestos)

”jitter” (los puntos se ven como alejados o inquietos) ”stack” (los puntos coincidentes son apilados, uno tras otro)

8) Crear la tabla de frecuencias completa

```

# frecuencias individuales
fab <- table(X);
fab # Frecuencia absoluta

## X
## 0  1  2  3  4  5
## 2  9 11  6  3  1

fre <- fab/length(X);
fre # frecuencias relativas

## X
##      0      1      2      3      4      5
## 0.06250 0.28125 0.34375 0.18750 0.09375 0.03125

Fac <- cumsum(fab);
Fac # frecuencias acumuladas

## 0  1  2  3  4  5
## 2 11 22 28 31 32

Far <- Fac/length(X);
Far # frecuencias acumuladas relativas

##      0      1      2      3      4      5
## 0.06250 0.34375 0.68750 0.87500 0.96875 1.00000

# tabla de frecuencias completa
options(digits = 2)
tabla <- data.frame(fab = fab, fre = fre, Fac = Fac, Far = Far)
names(tabla) <- c("X", "fab", "free.X", "fre", "Fac", "Far")
tabla

##   X fab free.X   fre Fac   Far
## 0 0   2     0 0.062   2 0.062
## 1 1   9     1 0.281  11 0.344
## 2 2  11     2 0.344  22 0.688
## 3 3   6     3 0.188  28 0.875
## 4 4   3     4 0.094  31 0.969
## 5 5   1     5 0.031  32 1.000

tfre <- data.frame(x = tabla$X, fab = tabla$fab, fre = tabla$fre,
                   Fac = tabla$Fac, Far = tabla$Far);
tfre

##   x fab   fre Fac   Far
## 1 0   2 0.062   2 0.062
## 2 1   9 0.281  11 0.344

```

```
## 3 2 11 0.344 22 0.688
## 4 3 6 0.188 28 0.875
## 5 4 3 0.094 31 0.969
## 6 5 1 0.031 32 1.000
```

Note que el cuadro resultante no tiene la presentación deseada para presentarla en un informe. Sin embargo, si estamos utilizando LATEX podemos utilizar la siguiente instrucción

```
xtable(tfre)

## Error in xtable(tfre): no se pudo encontrar la función "xtable"
```

y con esto nos genera el código correspondiente para incorporarlo en nuestro archivo.

9) Calcular los estadísticos descriptivos de la variable

```
# Estadísticos de tendencia central de los datos
media <- mean(X, na.rm = FALSE);
media

## [1] 2.1

# na.rm = FALSE, le indica a R que los datos
# faltantes son omitidos en el cálculo de la media.
for (i in 1:length(X)) {
  if(fab[i] == max(fab))
    break()
  moda <- names(fab[i]);
  moda # R no tiene incorporada una función para la moda
}
mediana <- median(X); mediana

## [1] 2
```

```
# Estadísticos de dispersión o variabilidad de los datos
range(X) # Devuelve el valor mínimo y máximo del conjunto de datos.

## [1] 0 5

cuasivar <- var(X);
cuasivar
```

```
## [1] 1.4

s <- sd(X);
s # Devuelve la cuasivarianza y la cuasivarianza muestral

## [1] 1.2

quantile(X, c(0.25, 0.5, 0.75))

## 25% 50% 75%
##    1    2    3

# Cálculo de Q1, Q2, Q3
quantile(X, 0.6)

## 60%
##    2

# En general se pueden encontrar cualquier percentil
# Conocer un resumen de los datos
resumen <- summary(X);
resumen # Min, Q1, Median, Mean, Q3, Max

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0     1.0     2.0     2.1     3.0     5.0

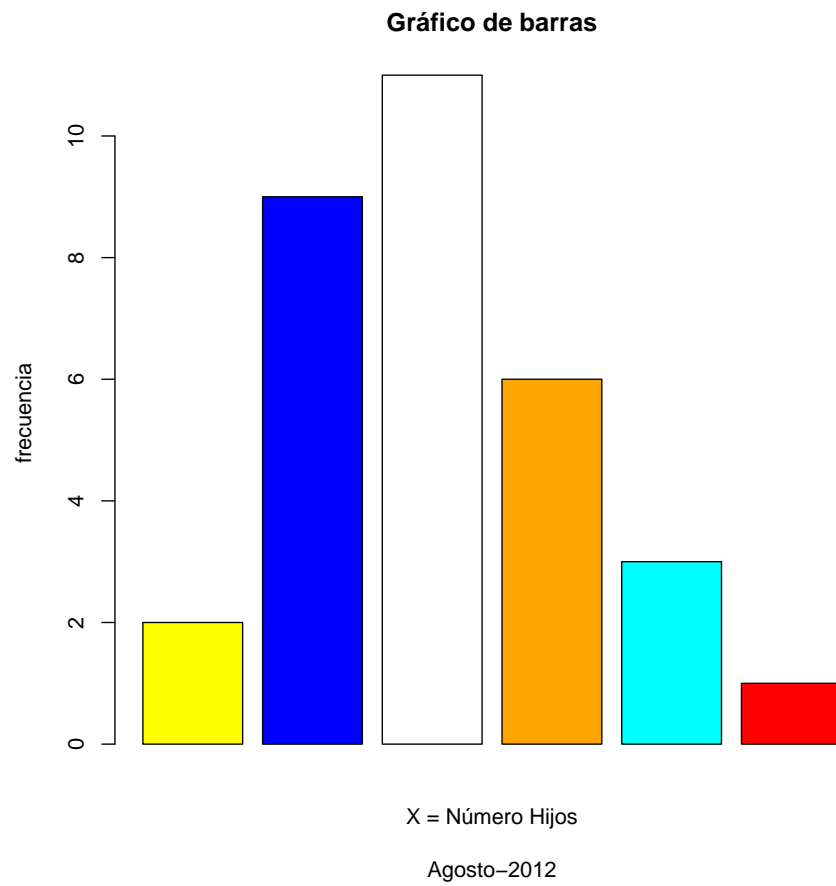
fivenum(X)

## [1] 0 1 2 3 5

# min, cuartil menor, mediana, cuartil mayor, max
```

10) Elaborar los gráficos que se le pueden aplicar a la variable discreta

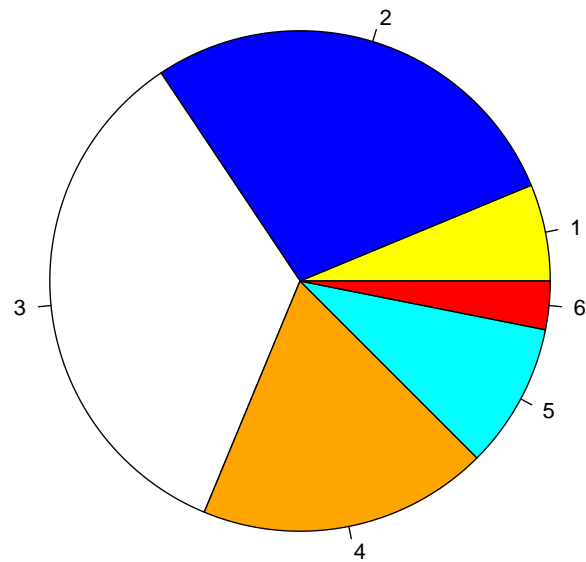
```
# Gráfico de barras (por ser pocos valores)
barplot(tfre[[2]], main = "Gráfico de barras", xlab = "X = Número Hijos\n",
        ylab = "frecuencia", col = c("yellow", "blue", "white", "orange", "cyan", "red"),
        sub = "Agosto-2012")
```



```
# Gráfico de pastel (por ser pocos valores)
pie(tfre[[2]], main = "Gráfico de pastel", xlab = "Número Hijos \n",
    col = c("yellow", "blue",
"white", "orange", "cyan", "red"), sub = "Agosto-2012")
```



Gráfico de pastel

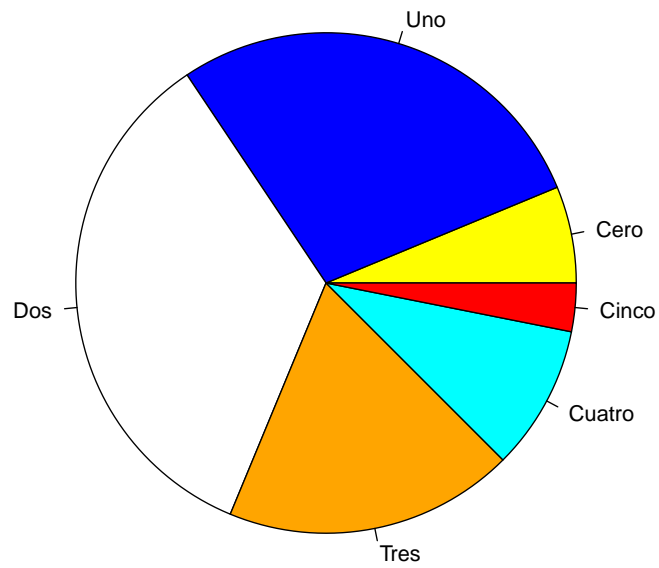


Número Hijos

Agosto-2012

```
# Se puede especificar nombres para las categorías
names(fab) = c("Cero", "Uno", "Dos", "Tres", "Cuatro", "Cinco")
pie(fab, main = "Gráfico de pastel", xlab = "X = Número Hijos\n",
    col = c("yellow", "blue",
            "white", "orange", "cyan", "red"), sub = "Agosto-2012")
```

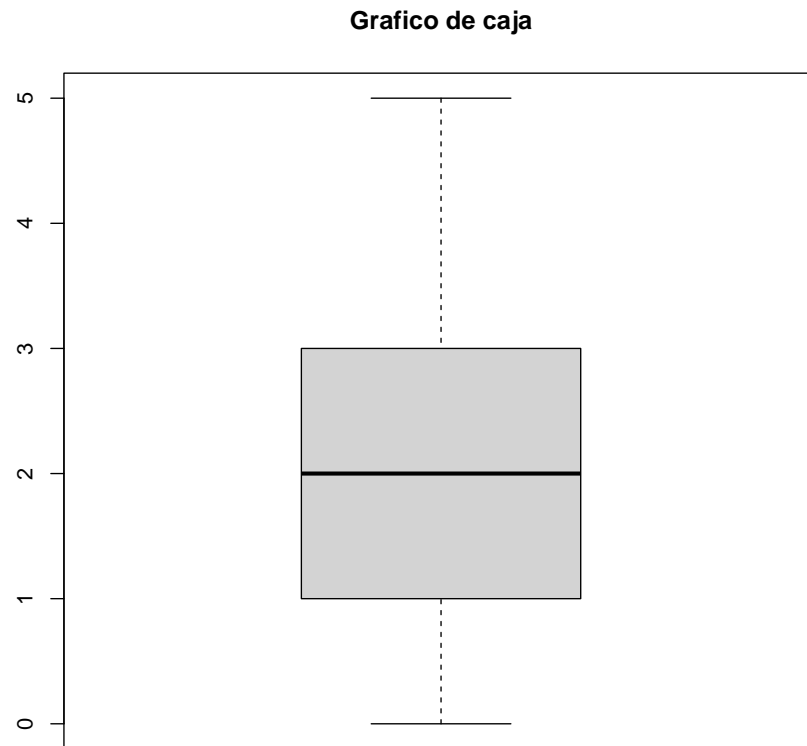
**Gráfico de pastel**



X = Número Hijos

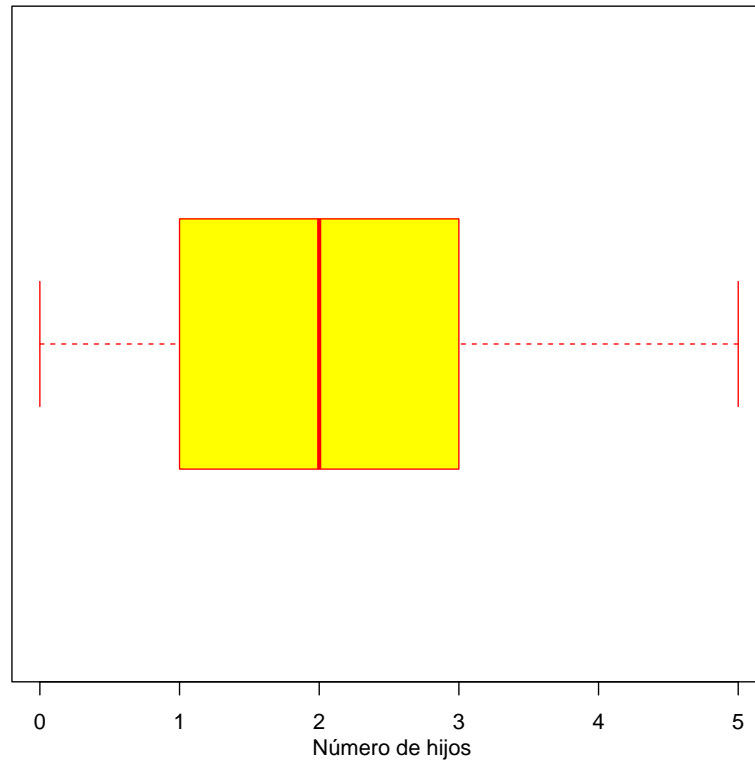
Agosto-2012

```
# Gráfico de cajas (box-plot) es la representación gráfica de los cinco números  
# Horizontal  
boxplot(X, main = "Grafico de caja", ylab = "Numero de hijos\n")
```



```
# Vertical  
boxplot(X, main="Gráfico de caja", xlab=" Número de hijos\n",  
        plot=TRUE, border="red", col="yellow", horizontal=TRUE)
```

**Gráfico de caja**



NOTE QUE TODOS LOS GRÁFICOS DE BARRAS Y DE PASTEL SON REALIZADOS APARTIR DE UNA TABLA DE FRECUENCIA, LA CUAL SE INDICA EN `tfre[[2]]`. TAMBIÉN SE PUDO UTILIZAR `tabla[[2]]`.