



Principios y Herramientas de Programación

Dra. Jessica Andrea Carballido

jac@cs.uns.edu.ar

```
4
5
names(sort(apply(ejemplo,1,sum)))[1]
] "d"
apply(ejemplo,1,sum)
b c d e f g h
6 5 4 5 6 7 8
sort(apply(ejemplo,1,sum))
c e b f a g h
5 5 6 6 7 7 8
sort(apply(ejemplo,1,sum))[1]
```

Dpto. de Ciencias e Ingeniería de la Computación

UNIVERSIDAD NACIONAL DEL SUR

Gráficos



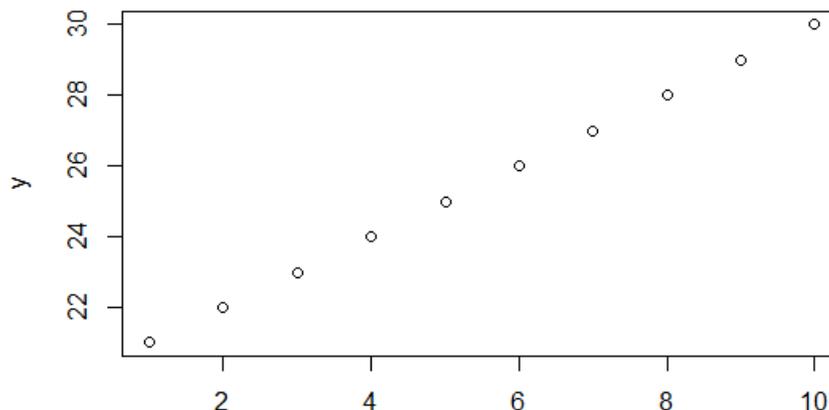
- Función general: **plot()**

Se puede usar para gráficos de dispersión, de funciones, de series de tiempo, etc.

El mayor poder se obtiene cuando se configura a través de sus **argumentos**.

plot(x, y), con x e y vectores de n coordenadas (no hay reciclado): obtenemos un gráfico de dispersión bivariado.

```
x=1:10  
y=21:30  
plot(x, y)
```



Dra. Jessica Andrea Carbal

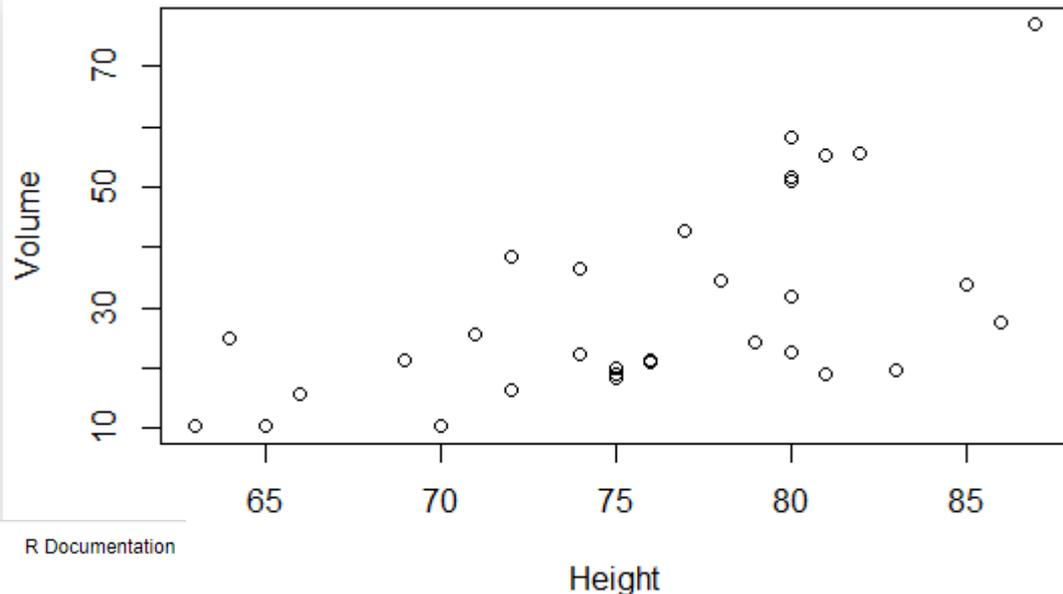
Los gráficos se pueden imprimir y guardar en distintos formatos

Gráficos

- Función general: **plot()**

- > attach(trees)
- > plot(Height, Volume)
- > dettach(trees)

Attach: guarda como variable global el nombre del dataset



trees {datasets} R Documentation

Diameter, Height and Volume for Black Cherry Trees

Description

This data set provides measurements of the diameter, height and volume of timber in 31 felled black cherry trees. Note that the diameter (in inches) is erroneously labelled Girth in the data. It is measured at 4 ft 6 in above the ground.

Usage

```
trees
```

Format

A data frame with 31 observations on 3 variables.

- [,1] Girth numeric Tree diameter (rather than girth, actually) in inches
- [,2] Height numeric Height in ft
- [,3] Volume numeric Volume of timber in cubic ft

```
> summary(trees)
```

Girth		Height		Volume	
Min.	: 8.30	Min.	:63	Min.	:10.20
1st Qu.:	:11.05	1st Qu.:	:72	1st Qu.:	:19.40
Median	:12.90	Median	:76	Median	:24.20
Mean	:13.25	Mean	:76	Mean	:30.17
3rd Qu.:	:15.25	3rd Qu.:	:80	3rd Qu.:	:37.30
Max.	:20.60	Max.	:87	Max.	:77.00

```
Max. : 50.00 Max. : 87 Max. : 77.00
```

Gráficos



- PLOT: También se pueden incluir [expresiones](#) en los parámetros correspondientes a los ejes. Por ejemplo:

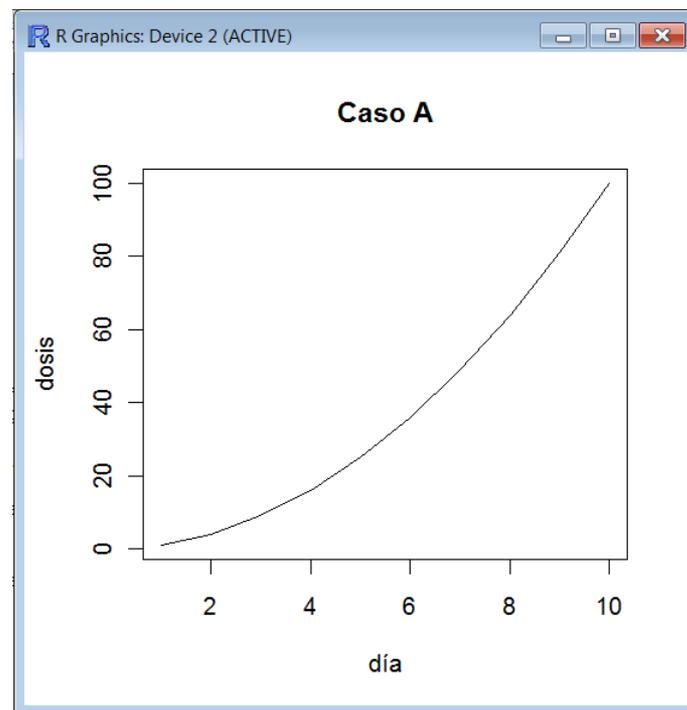
```
> x <- seq(1:10)
```

```
> plot(x, x^2, type="l", main="Caso A", xlab="día", ylab="dosis")
```

```
Ídem a hacer: curve(x^2, 1,10,...)
```

curve tiene por defecto `type="l"`.
plot tiene por defecto `type="p"`

En *plot*, `x` debe existir.
En *curve*, `x` (objeto "virtual") se asocia a los valores de los argumentos o variará entre 0 y 1 por defecto.



Gráficos



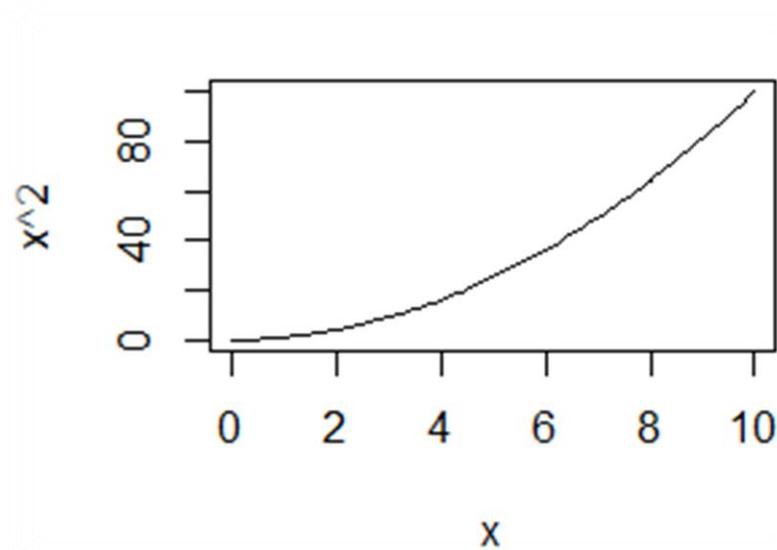
- Funciones continuas: **curve()**

La forma básica de llamada es:

curve(expression, from, to)

La expresión debe escribirse
en función de x .

Ejemplo: `curve(x^2, 0, 10)`



Gráficos



- Funciones continuas: **curve()**

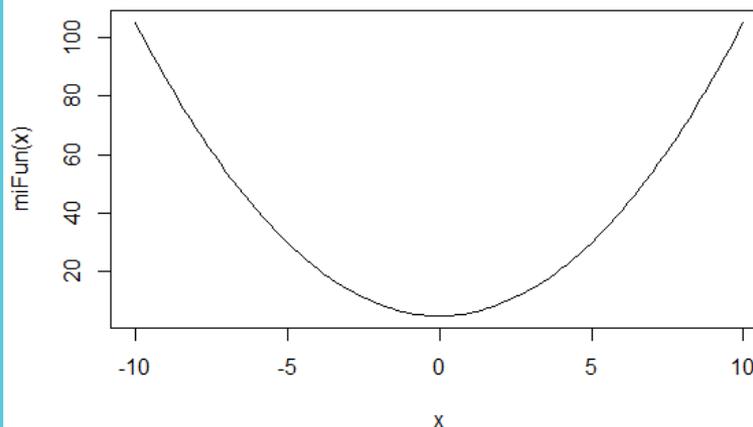
La forma básica de llamada es:

curve(expresion, from, to)

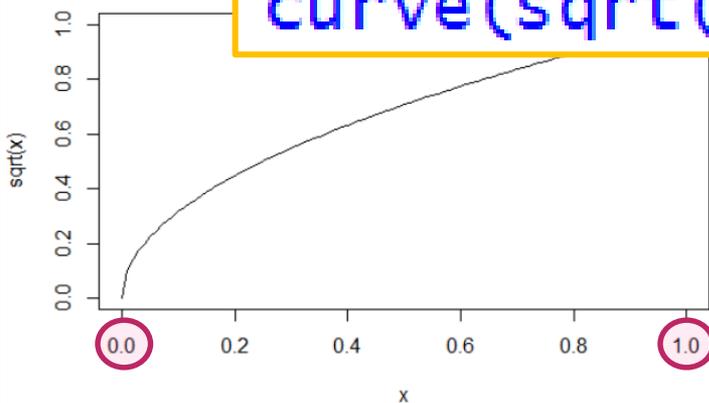
La expresión debe escribirse en función de x .

La expresión puede ser una invocación a una función.

```
miFun=function(x){(x^2+5)}  
curve(miFun(x), -10, 10)
```

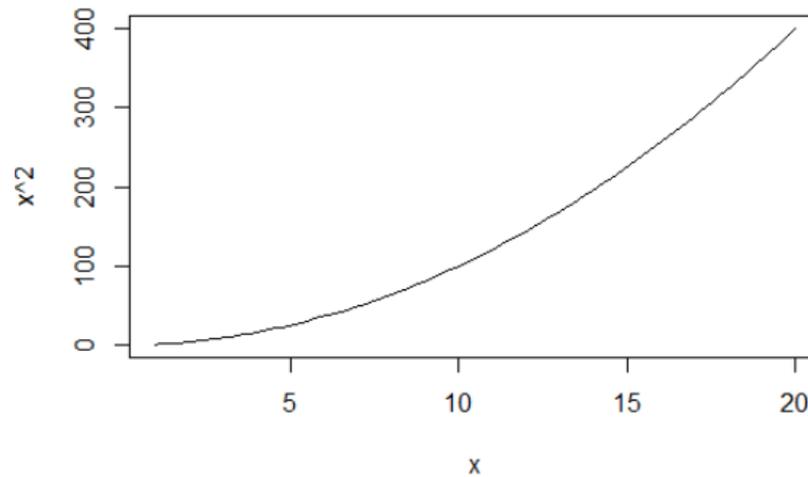


```
curve(sqrt(x))
```



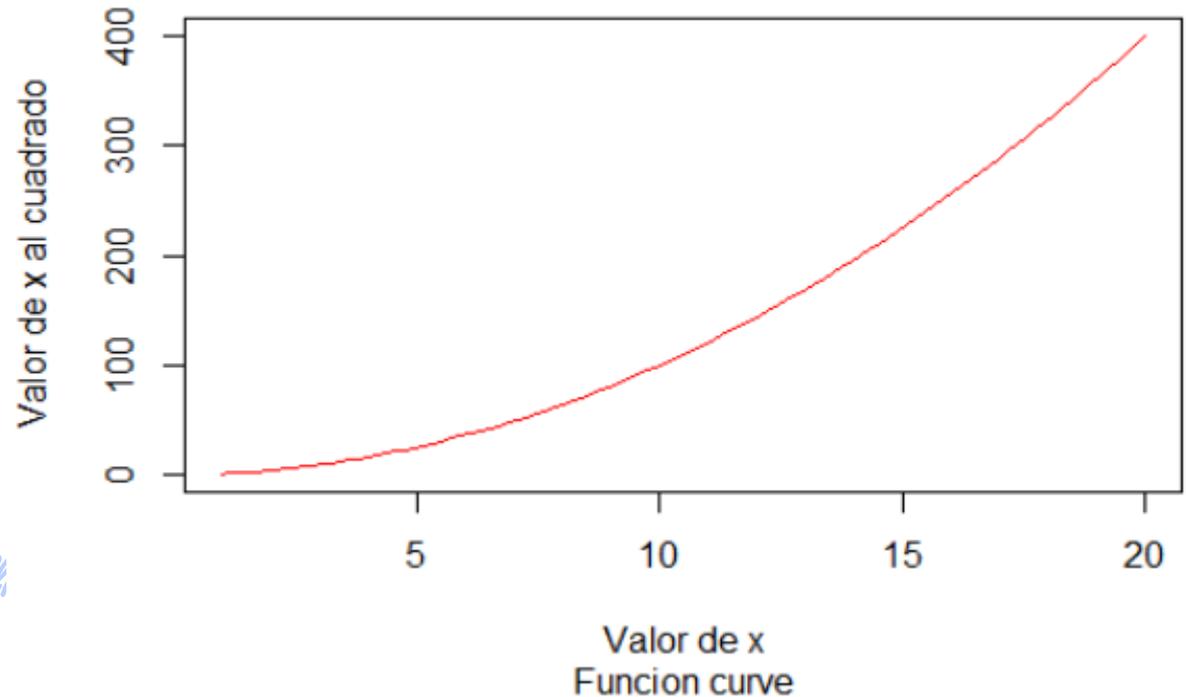
x puede omitirse en la función:
`curve(sqrt)`

```
> curve(x^2,1,20)
```



```
> curve(x^2,1,20,xlab="Valor de x", ylab="Valor de x al cuadrado", main="GRAFICO", col="red", sub="Funcion curve")
```

GRAFICO



Dra. Jessica Andrea Carballi
CONICET - DCIC (UNS)

Gráficos



Algunos **parámetros** de funciones gráficas de alto nivel:

add= FALSE (defecto): Si es TRUE superpone el gráfico en el ya existente.

axes= TRUE (defecto): si es FALSE no dibuja los ejes ni la caja del gráfico.

type="p" (defecto): especifica el tipo de gráfico por ejemplo p= puntos, l= líneas, b= puntos conectados por líneas, h= líneas verticales.

xlim=, ylim=: especifica los límites inferiores y superiores de los ejes; por ejemplo con **xlim=c(1,5)** o **xlim=range(x)**

xlab=" ", ylab=" ": añade los títulos a los ejes.

main=" ": añade el título principal.

sub=" ": añade un subtítulo(letra más pequeña)

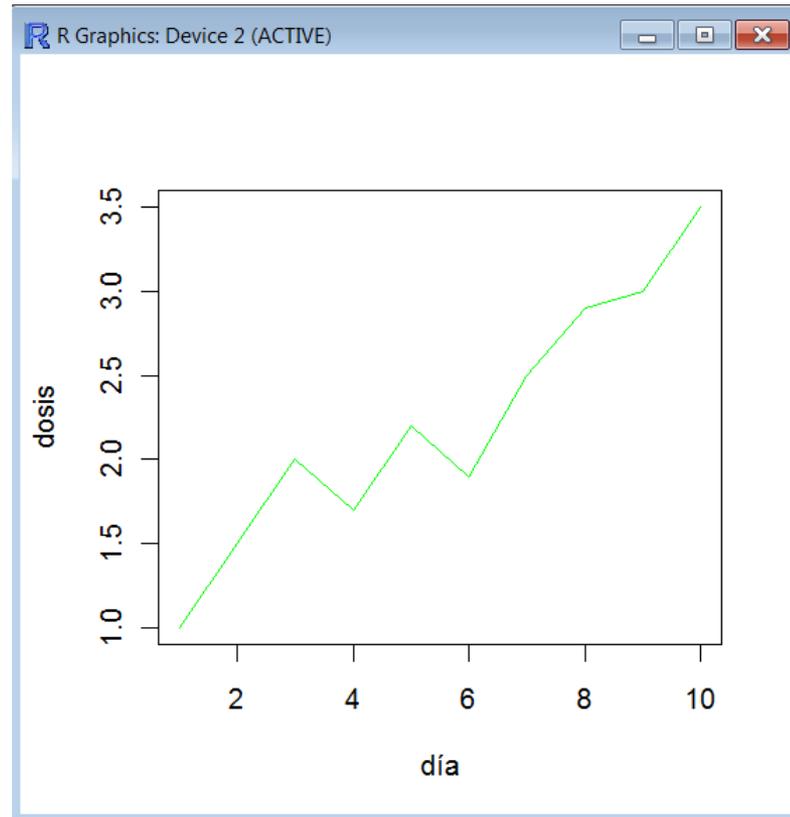
col= elegimos el color que deseamos para cada objeto.



Gráficos



- **col:** para indicar el color que se utilizará para graficar.
- ```
> plot(x, y, type="l", xlab="día", ylab="dosis", col="green")
```

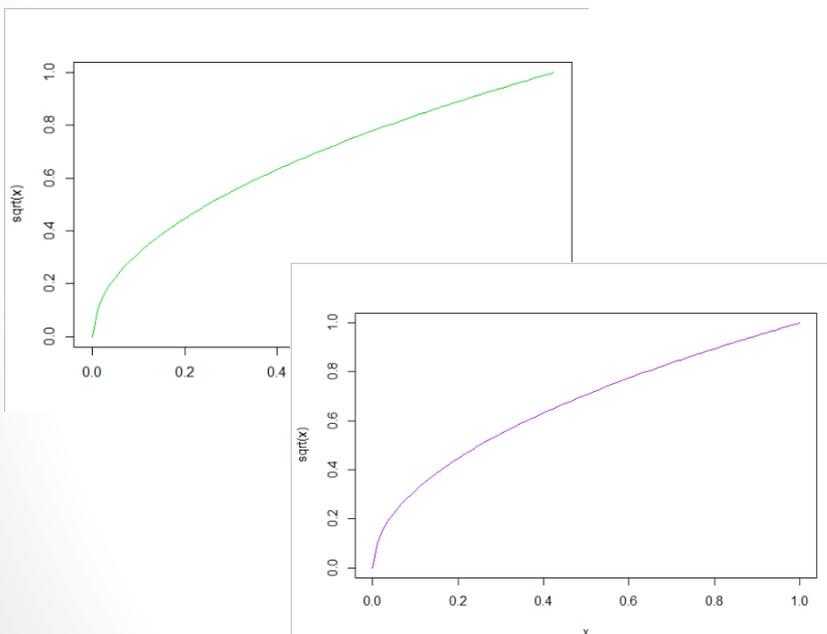


# Gráficos



- **col:** para indicar el color que se utilizará para graficar. Puede indicarse con un número.

```
> curve(sqrt, col=115)
```



- Paleta por defecto

1: black

2: red

...

115: verde

- Paleta colors()

1: white

2: aliceblue

...

115: violeta

```
palette(colors()) # seteo la paleta
palette("default") # seteo la default
palette() # consulto cual es la actual
```

Ver archivo Rcolor.pdf

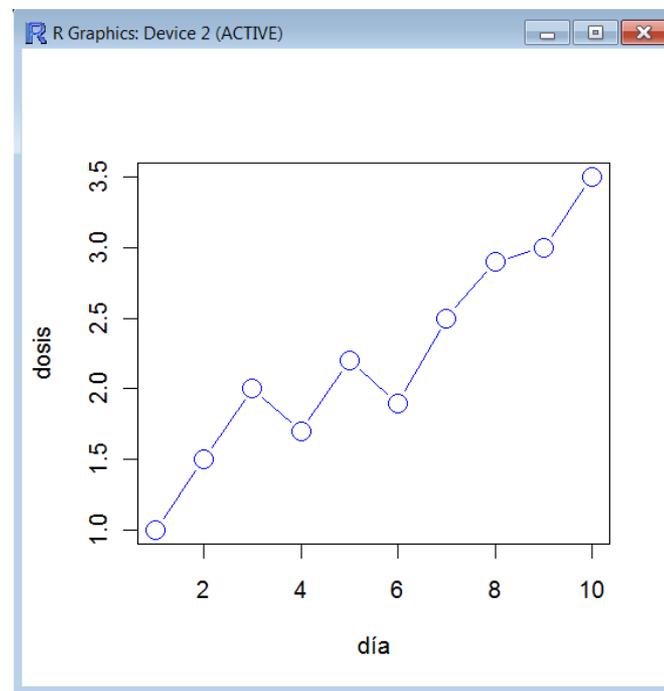


# Gráficos



- **cex**: un valor numérico que permite magnificar o reducir los símbolos en forma relativa a las dimensiones por defecto.
- **cex.axis**, **cex.lab** y **cex.main** son similares pero aplicados a las leyendas y títulos del gráfico.

```
> plot(x, y, type="b", xlab="día", ylab="dosis", cex=1.9, col="blue")
```



# Gráficos



- Funciones de “bajo nivel” para agregar elementos a gráficos existentes:

| Función                 | Operación                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------|
| <code>abline()</code>   | Agrega una línea con ordenada al origen y pendiente especificadas                       |
| <code>arrows()</code>   | Agrega una flecha en coordenadas de comienzo y final especificadas                      |
| <code>lines()</code>    | Agrega líneas entre coordenadas                                                         |
| <code>points()</code>   | Agrega puntos en coordenadas especificadas                                              |
| <code>rug()</code>      | Agrega, sobre el eje x, los datos del eje x como pequeñas líneas verticales             |
| <code>segments()</code> | Agrega un segmento lineal entre un par de puntos                                        |
| <code>text()</code>     | Agrega un texto en coordenadas establecidas, posiblemente dentro de la región graficada |
| <code>title()</code>    | Agrega títulos, subtítulos, etc.                                                        |



# Gráficos



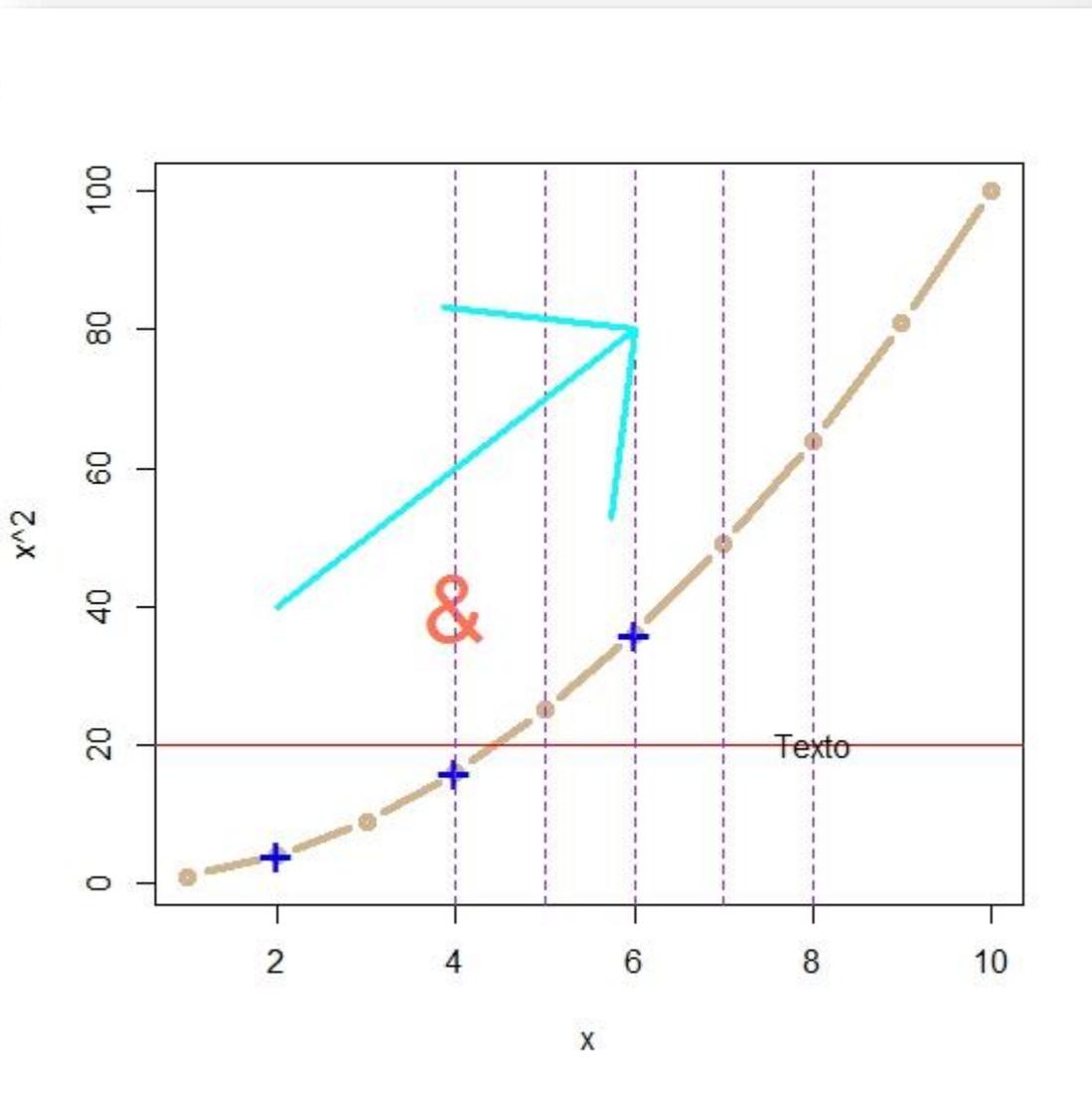
```
x <- seq(1:10)
plot(x,x^2,type="b", lwd=4, col="tan") # lwd aumenta el ancho de linea
points(c(2,4,6), c(4,16,36), pch="+", cex=2, col="blue")
points(4,40,pch="&",col="coral1",cex=3)
abline(h = 20, col = "red")
abline(v = 4:8, col="purple", lty=2) # lty indica el tipo de linea
arrows(2, 40, 6, 80, lwd=3, length=1, angle=45, col="cyan")
text(8,20,"Texto") # centrado en ese punto
```



# Gráficos



```
x <- seq(1:10)
plot(x,x^2,type="b", lwd=
 linea
points(c(2,4,6), c(4,16,36)
points(4,40,pch="&",col=
abline(h = 20, col = "red"
abline(v = 4:8, col="purp
arrows(2, 40, 6, 80, lwd=
text(8,20,"Texto") # cent
```



*Dra. Jessica Andrea Carballido*  
CONICET - DCIC (UNS)

# Gráficos



Cambiar los parámetros **gráficos de manera permanente:**

## Función *par()*

Los parámetros establecidos con esta función quedarán definidos para el resto de los gráficos que se realicen.

Varios gráficos en una ventana: ***mfrow()*** y ***mfcop()***: reciben un vector  $c(m,n)$  y arman una grilla de  $m \times n$  para ir llenando con los gráficos, por fila o por columna respectivamente. Se establece como un valor de parámetro.

Color del fondo: **bg**

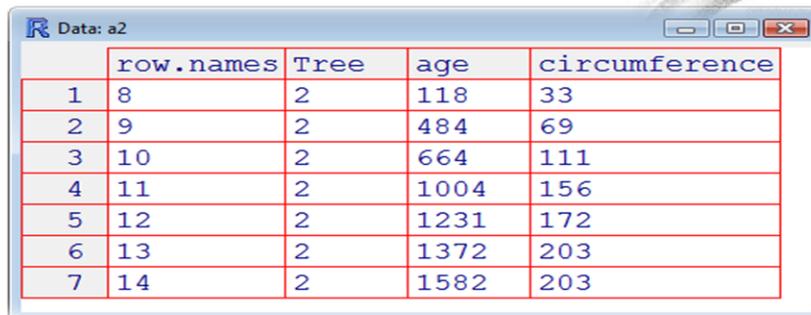
Color grafico, de títulos y leyendas (**col, col.main, col.lab**)

Márgenes (**mar, oma**)



# Gráficos

```
> data(Orange)
> attach(Orange)
> a1=Orange[Tree==1,]
> a2=Orange[Tree==2,]
> a3=Orange[Tree==3,]
> a4=Orange[Tree==4,]
> View(a2)|
> |
```



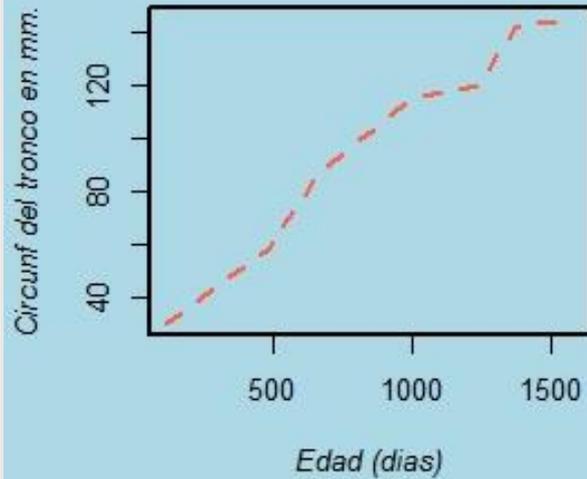
|   | row.names | Tree | age  | circumference |
|---|-----------|------|------|---------------|
| 1 | 8         | 2    | 118  | 33            |
| 2 | 9         | 2    | 484  | 69            |
| 3 | 10        | 2    | 664  | 111           |
| 4 | 11        | 2    | 1004 | 156           |
| 5 | 12        | 2    | 1231 | 172           |
| 6 | 13        | 2    | 1372 | 203           |
| 7 | 14        | 2    | 1582 | 203           |

← # idem a4=subset(Orange, Tree==4)

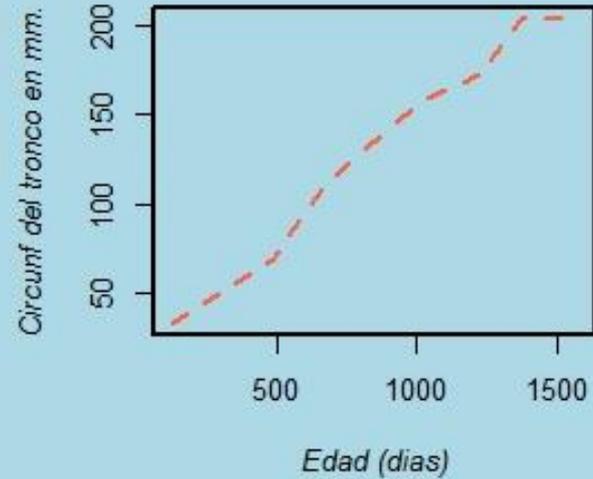
- `par(bg="lightblue", mfrow=c(2,2), col.main="blue", cex.main=2, lty=2, lwd=2, font.lab=3, font.main=2)`
- `plot(a1$age, a1$circumference, main="Arbol N 1", type="l", col="tomato", cex=3, xlab="Edad (dias)", ylab="Circunf del tronco en mm.")`
- `plot(a2$age, a2$circumference, main="Arbol N 2", type="l", col="tomato", cex=3, xlab="Edad (dias)", ylab="Circunf del tronco en mm.")`
- `plot(a3$age, a3$circumference, main="Arbol N 3", type="l", col="tomato", cex=3, xlab="Edad (dias)", ylab="Circunf del tronco en mm.")`
- `plot(a4$age, a4$circumference, main="Arbol N 4", type="l", col="tomato", cex=3, xlab="Edad (dias)", ylab="Circunf del tronco en mm.", col.main="red")`



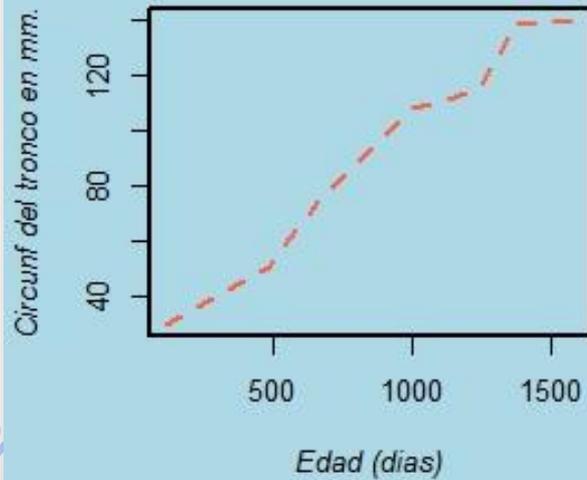
## Arbol N 1



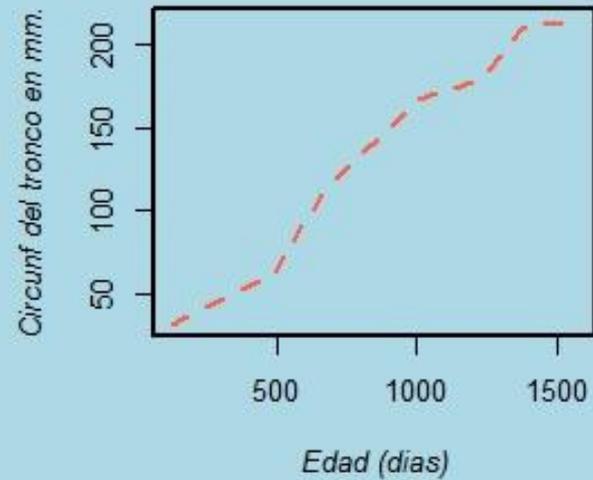
## Arbol N 2



## Arbol N 3



## Arbol N 4

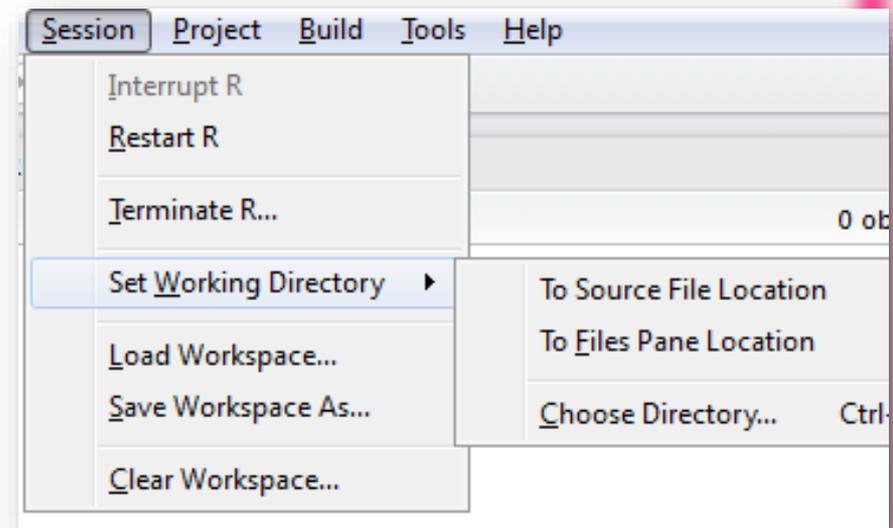


# Guardando gráficos en archivos



- > pdf("grafico1.pdf") # abre el archivo para escritura  
en el directorio que tiene seteado por defecto
- > plot(x, y, main="Plot a pdf")
- > dev.off() # cierra el último dispositivo gráfico abierto

El *path* puede modificarse desde el entorno  
o con el comando `setwd("path")`.



Dra. Jessica Andrea Carballido  
CONICET - DCIC (UNS)



# Guardando gráficos en archivos



```
> pdf("graficoNuevo.pdf") ←
> par(bg="pink")
> plot(x,y,xlim=c(0,10))
> par(bg="green")
> plot(x,y,xlim=c(0,10))
> attach(Orange)
> a1=Orange[Tree==1,]
> a2=Orange[Tree==2,]
> a3=Orange[Tree==3,]
> a4=Orange[Tree==4,]
> par(bg="lightblue", mfrow=c(2,2), col.main="blue", cex.main=2, lty=2, lwd=2,
font.lab=3, font.main=2)
> plot(a1$age, a1$circumference, main="Arbol N 1", type="l", col="tomato", cex=3,
xlab="Edad (dias)", ylab="Circunf del tronco en mm.")
> plot(a2$age, a2$circumference, main="Arbol N 2", type="l", col="tomato", cex=3,
xlab="Edad (dias)", ylab="Circunf del tronco en mm.")
> plot(a3$age, a3$circumference, main="Arbol N 3", type="l", col="tomato", cex=3,
xlab="Edad (dias)", ylab="Circunf del tronco en mm.")
> plot(a4$age, a4$circumference, main="Arbol N 4", type="l", col="tomato", cex=3,
xlab="Edad (dias)", ylab="Circunf del tronco en mm.", col.main="red")
> dev.off() ←
```



# Guardando gráficos en archivos



The screenshot shows Adobe Reader displaying a PDF document titled 'graficoNuevo.pdf'. The document contains four graphs arranged in a 2x2 grid. The first two graphs are highlighted with a pink and a green background, respectively. The last two graphs are titled 'Arbol N 1', 'Arbol N 2', 'Arbol N 3', and 'Arbol N 4'. Each graph plots 'Circunf del tronco en mm.' (Trunk circumference in mm) on the y-axis against 'Edad (dias)' (Age in days) on the x-axis. The x-axis for all graphs ranges from 0 to 1500 days, and the y-axis ranges from 0 to 30 mm for the first two graphs and 40 to 140 mm for the last two graphs. The graphs show a sigmoidal growth curve for each tree.

**Arbol N 1**

**Arbol N 2**

**Arbol N 3**

**Arbol N 4**



# Gráficos 3D

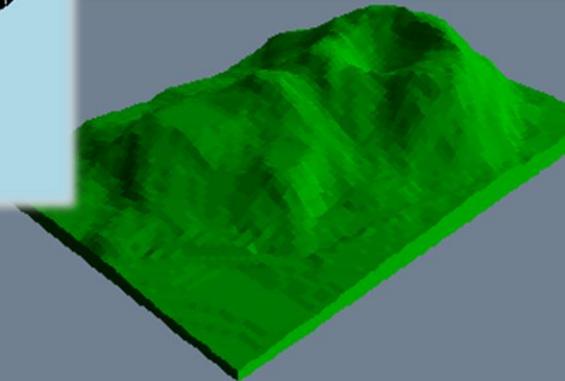
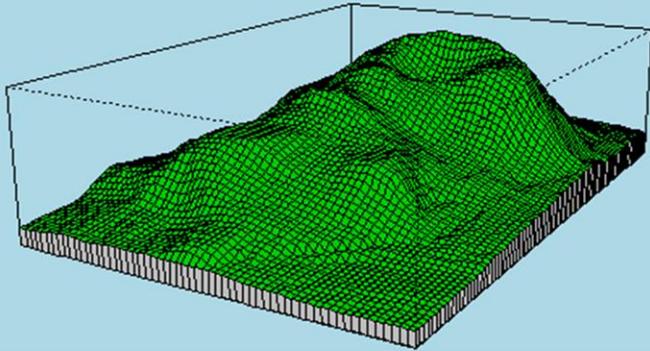


- **persp**(x, y, z) permite hacer gráficos en perspectiva.

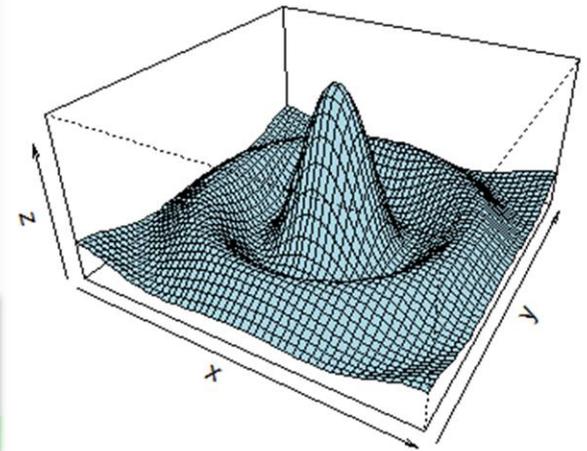
Parámetros: límites de los ejes, título, subtítulo, etiquetas de los ejes, ángulos del gráfico, etc.

> demo(persp)

*Maunga Whau  
One of 50 Volcanoes in the Auckland Region.*



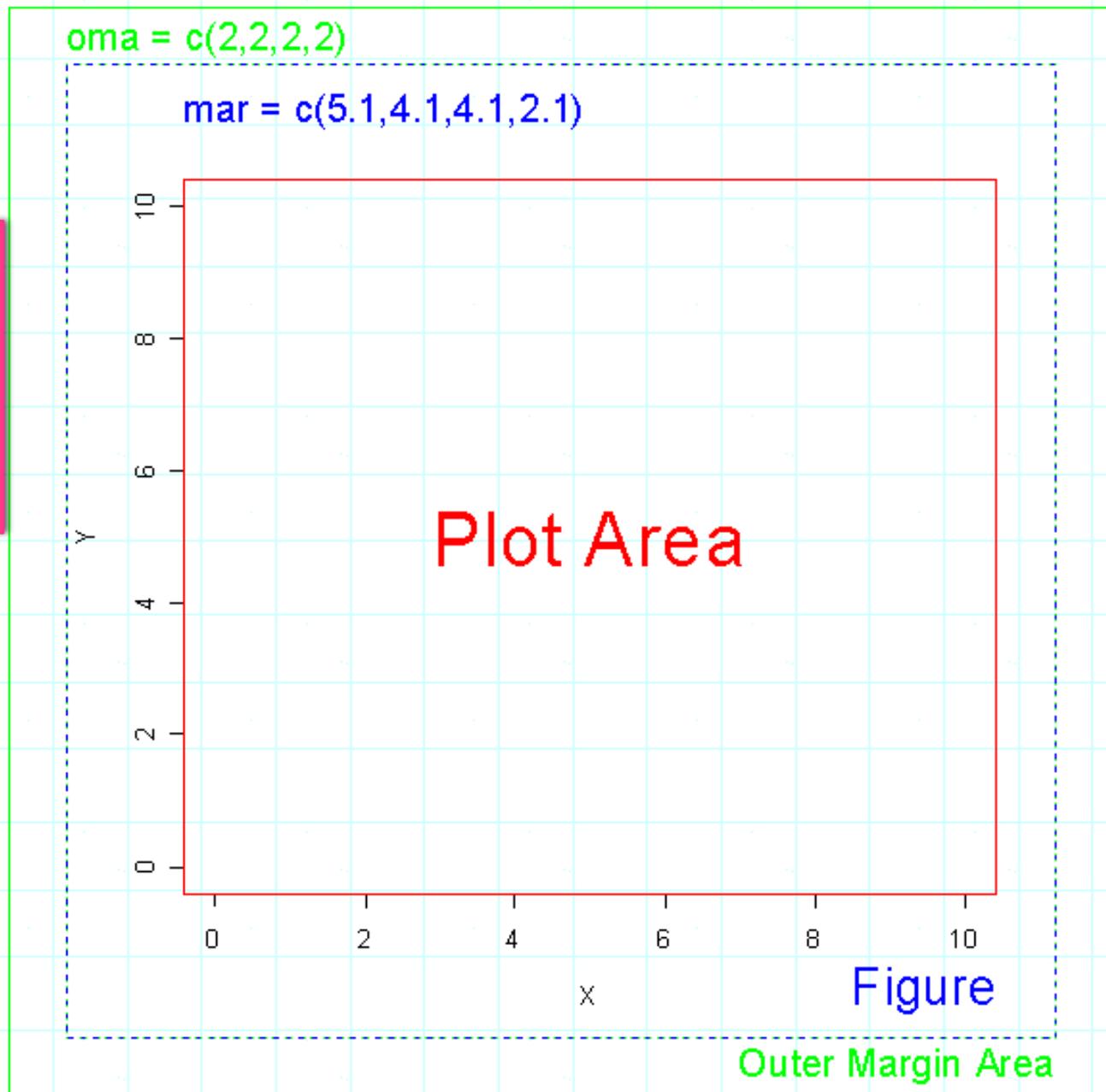
$$z = \text{Sinc}(\sqrt{x^2 + y^2})$$



# Márgenes

## Referencia

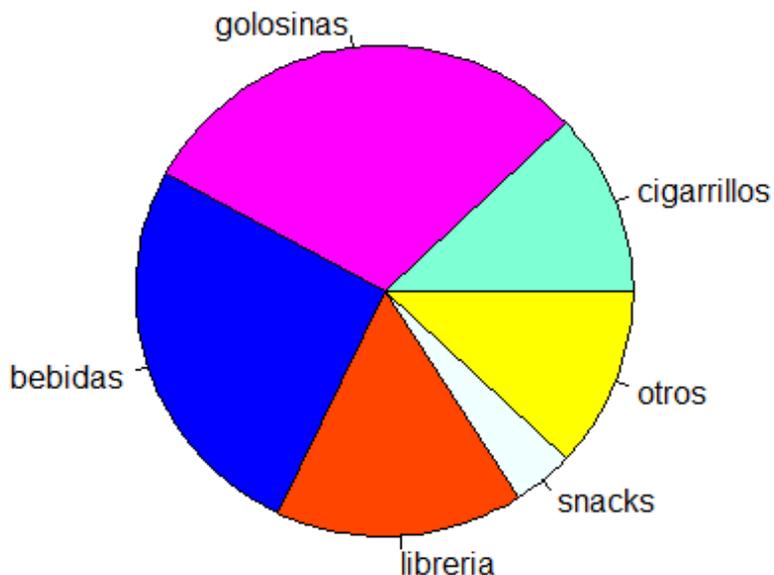
- 1ro: inferior
- 2do: izquierdo
- 3ro: superior
- 4to: derecho



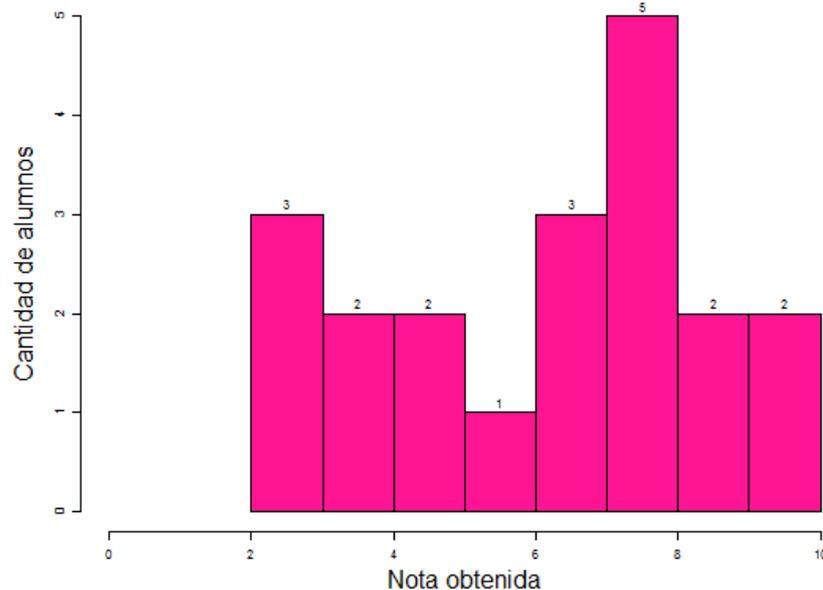
```
> ventas=c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
>
> names(ventas)=c("cigarrillos", "golosinas", "bebidas", "libreria", "snacks", "otros")
> pie(ventas, col=c(9,450,26,503,13,652), main="Ventas del kiosco")
>
```

Si no tenía nombres asignados, podemos usar el parámetro *labels*

**Ventas del Kiosco**



**Histograma de resultados:  
Nota entre 0 y 10**



```
7 notas=round(runif(20, 2, 10))
8 par(mar=c(6, 6, 5, 0.5))
9 hist(notas, freq = T, border="black", xlim=c(0,10),
10 labels=T, col = "deeppink",
11 xlab = "Nota obtenida", ylab= "Cantidad de alumnos", cex.lab=2,
12 main = "Histograma de resultados: \n Nota entre 0 y 10", cex.main=2.5)
13 #notas
14 #[1] 4 9 8 6 7 8 7 2 8 5 8 3 4 7 5 2 10 10 8 9
15 # como tiene el parametro right=TRUE por defecto, los intervalos son (a,b] salvo el primero.
```

# Guardando y Recuperando Objetos



- Todo objeto generado en el ambiente de trabajo en R puede ser guardado en un archivo de tipo RData usando la instrucción **save**.
- Para recuperar dicho objeto en sesiones futuras de R debemos utilizar la instrucción **load**.
- Por ejemplo:
  - > `save(v, "v.RData")` # guarda el objeto v en el archivo v.RData.
  - > `load("v.RData")` # carga en el ambiente de trabajo al objeto v salvado en v.RData.

Un archivo RData puede almacenar más de un objeto de R.  
Por ejemplo, puedo hacer `save(v1, v2, "v.RData")`  
y guardo tanto a v1 como a v2.

## PARÁMETROS GRÁFICOS (FUNCIÓN PAR)

Además de la utilización de los gráficos de bajo nivel, la presentación de estos puede verse mejorada con parámetros gráficos adicionales. Estos se pueden utilizar como opciones aunque no funcionan para todas. Este procedimiento se puede realizar también utilizando la función `par()` que nos permite cambiar de forma permanente los parámetros gráficos. Veamos las principales posibilidades que nos ofrece esta función.

- `adj`: El valor que tome determina la forma en la cual las “cadenas” de texto son justificadas. 0 para justificar el texto a la izquierda, 0.5 para centrar el texto y 1 para justificarlo a la derecha.
- `ann`: Si lo ajustamos como FALSE entonces en las funciones de graficación de alto nivel no se realiza anotación sobre los ejes, sólo producen los ejes con sus títulos y el respectivo gráfico.
- `ask`: Si es TRUE, el usuario es preguntado por el imput antes que una nueva figura sea dibujada.
- `bg`: Nos especifica que color va a ser utilizado para los fondos de los gráficos.
- `bty`: Determina el tipo de caja que será dibujada alrededor de los gráficos “o”, “l”, “7”, “c”, “u”, “j”, “n”(suprime la caja)
- `cex`: Es un valor que controla el tamaño del texto y símbolos con respecto al valor por defecto. Podemos utilizar, `cex.axis` (números de los ejes), `cex.lab` (títulos de los ejes), `cex.main` (título principal) y `cex.sub` (subtítulo)
- `cin`: Tamaño del carácter “(ancho, alto)” en pulgadas
- `col`: controla el color de los símbolos al igual que podemos elegir el color de cada elemento del gráfico con los comandos: `col.axis` (ejes), `col.lab` (títulos de los ejes), `col.main`(título principal) y `col.sub` (subtítulo)
- `fg`: El color a ser usado para el primer plano de los gráficos. Es el color que por defecto es usado en objetos como los ejes y las cajas alrededor de los gráficos.
- `font`: Un entero que especifica la fuente a usar para el texto, de modo que 1 corresponde a texto plano, 2 corresponde a texto en negrita, 3 texto en itálica, 4 texto itálica-negrita. Podemos utilizar, `font.axis`, `font.lab`, `font.main`, `font.sub`.



- las: Un número (0, 1, 2, 3), con el cual se especifica el estilo de las etiquetas de las anotaciones de los ejes: 0 (defecto) paralelo al eje, 1 horizontal, 2 perpendicular al eje, 3 vertical
- lty: Un entero que controla el tipo de las líneas (1: sólida, 2: quebrada, 3: punteada, 4: punto-línea, 5: línea larga-corta, 6: dos líneas cortas)
- lwd: Un número positivo para determinar el ancho de línea (defecto=1)
- mar: Un vector con 4 valores numéricos que controla el espacio entre los ejes y el borde de la gráfica en la forma  $c(\text{inferior}, \text{izquierda}, \text{superior}, \text{derecha})$  los valores por defecto son  $c(5,4,4,2) + 0.1$
- mfcol: Es un vector del tipo  $c(nf,nc)$  que divide la ventana gráfica como una matriz  $cn$   $nf$  filas y  $nc$  columnas. Las gráficas se van dibujando sucesivamente por columnas.
- mfrow: igual que el anterior pero las gráficas se dibujan por filas.
- pch: controla el tipo de símbolo con un entero entre 1 y 25.
- ps: Un entero que controla el tamaño (en puntos) de texto y símbolos.
- pty: Caracter entre comillas para especificar el tipo de región gráfica a ser usada: "s" región cuadrada, "m" región máxima
- xaxt: si  $xaxt = "n"$  el eje x se coloca pero no se muestra
- yaxt: si  $yaxt = "n"$  el eje y se coloca pero no se muestra

[https://www.datanalytics.com/libro\\_r/introduccion-a-ggplot2.html](https://www.datanalytics.com/libro_r/introduccion-a-ggplot2.html)



*Dra. Jessica Andrea Carballido*

*CONICET - DCIC (UNS)*





¡¡Que tengas un bonito dia!!