

# INTRODUCCIÓN A LA REPRESENTACIÓN GRÁFICA

Ramon Ceballos

18/1/2021

## 1. Uso de la función `plot()`

### Gráfico básico de puntos

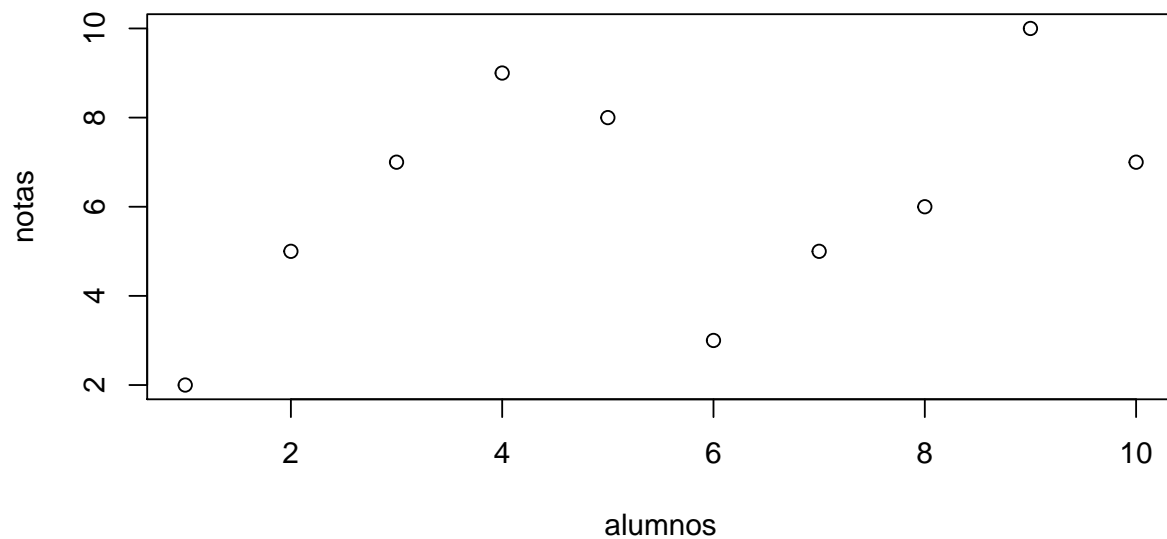
Puede ser un gráfico de dispersión (puntos) o gráfico continuo (funciones).

Para hacer gráficos de puntos se utiliza:

- `plot(x,y)`: para dibujar un gráfico básico de puntos siendo  $x, y$  vectores numéricos
  - `plot(x) = plot(1:length(x),x)`
- `plot(x,función)`: para dibujar el gráfico de una función

### Ejemplos

```
alumnos = c(1:10)
notas = c(2,5,7,9,8,3,5,6,10,7)
plot(alumnos,notas)
```



```
x = c(2,6,4,9,-1)
y = c(1,8,4,-2,4)
plot(x,y)
```

Si no incorporamos vector `y`, R nos va a tomar el parámetro `x` como si fuese el vector de datos `y` : `plot(1:n, x)`. Es decir, “x” toma el valor del índice del valor dentro del vector e “y” el valor como tal.

```
plot(2^(1:6))
```

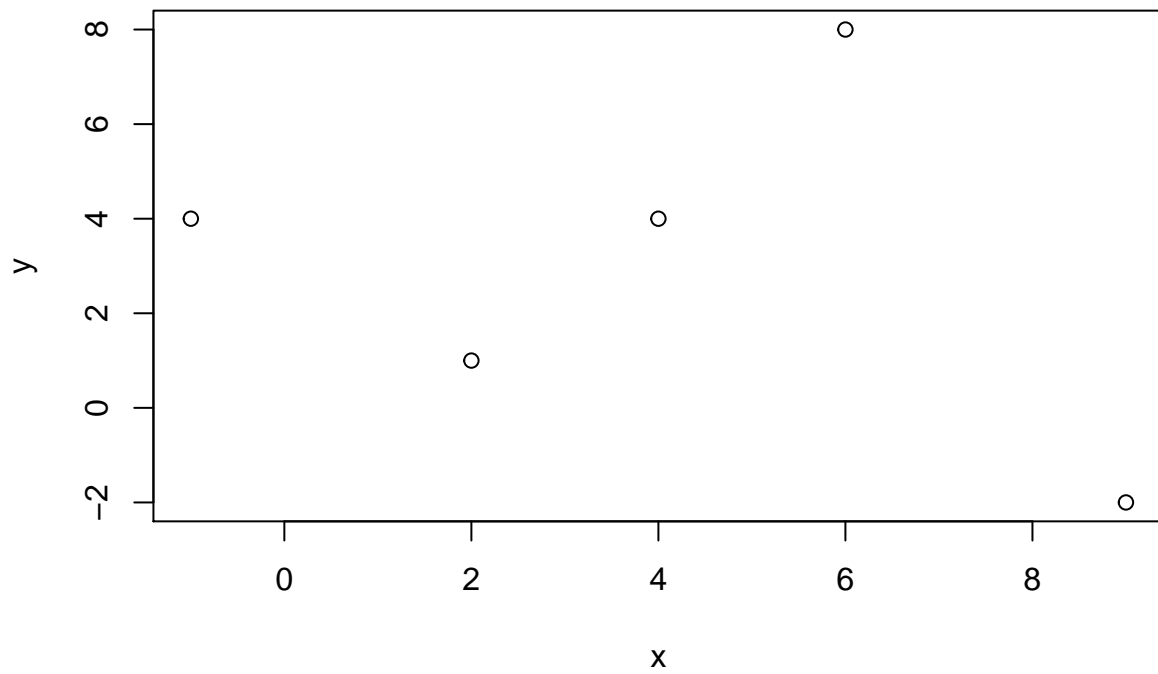
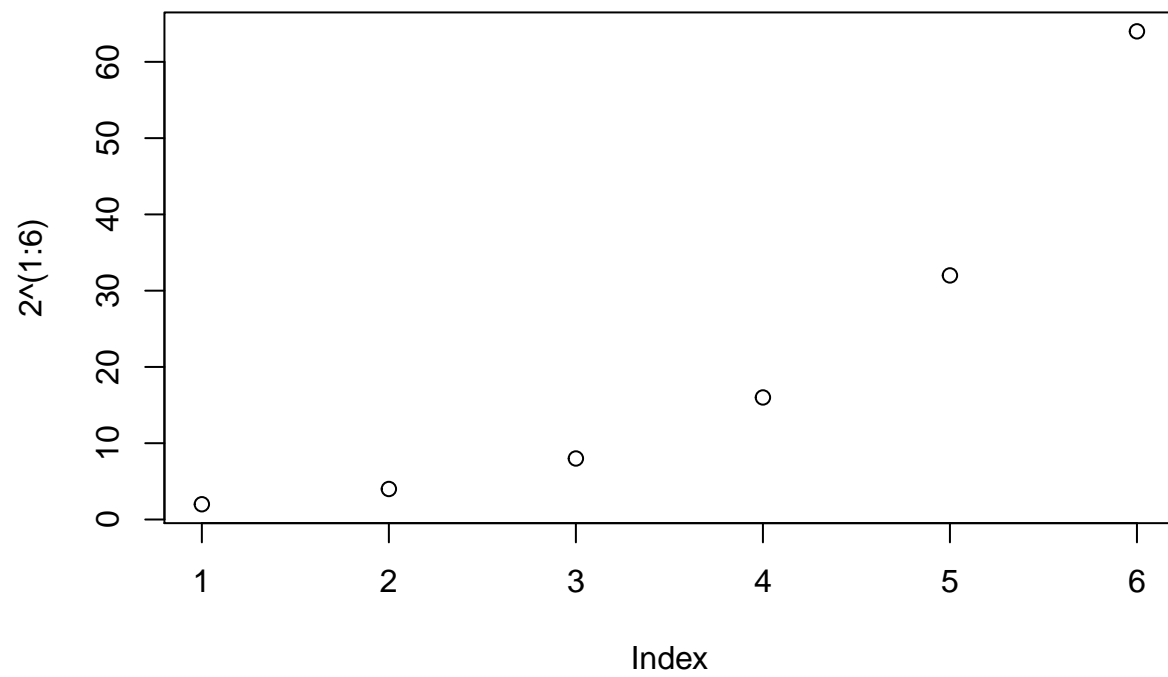
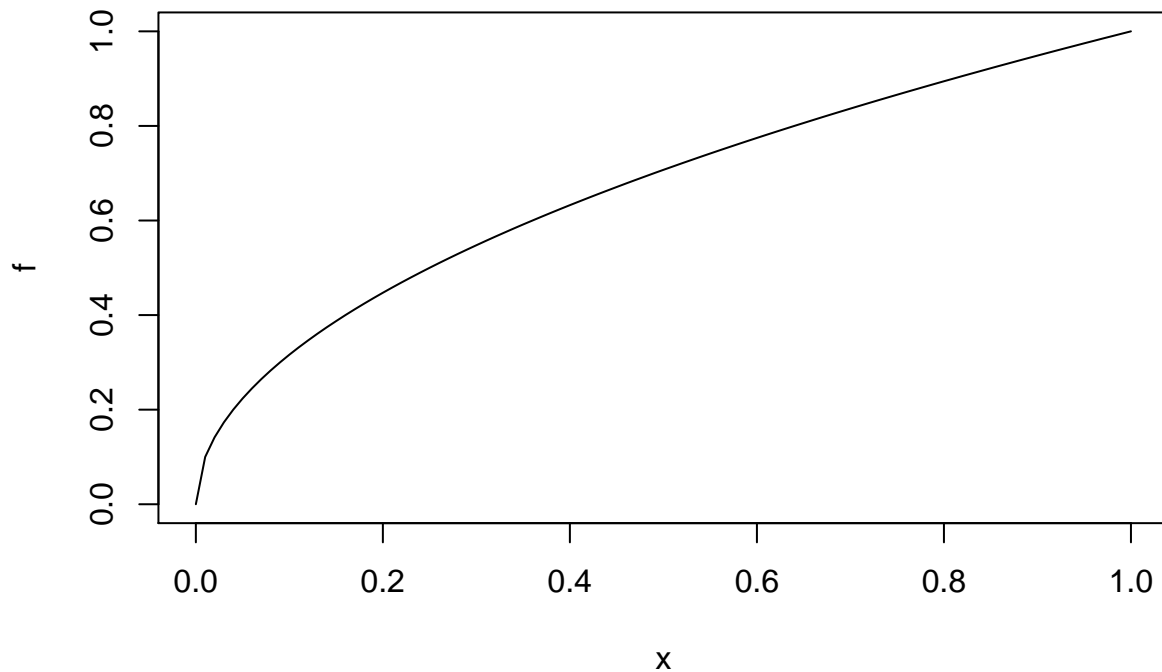


Figure 1: GRAFICO DE EJEMPLO NÚMERO 1. Uso de fig.cap



Si queremos representar una función  $f(x)$ :

```
f <- function(x){ sqrt(x) }  
plot(f)
```



Por defecto el plot de la gráfica lo realiza entre 0 y 1.

## 2. Parámetros de la función `plot()`

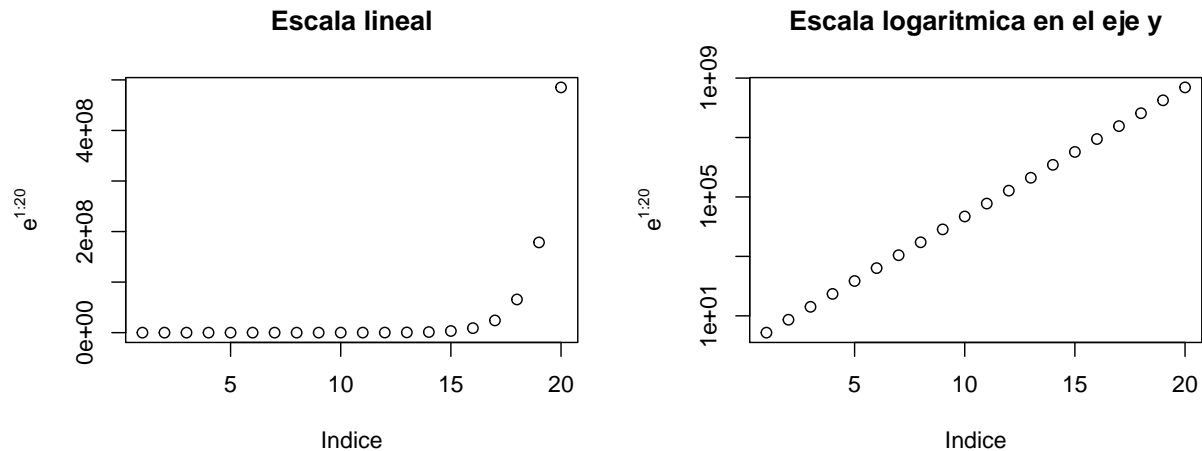
### 2.1 Primera clase

Algunos parámetros utilizados para modificar los parámetros de visualización de un gráfico son:

- `log`: para indicar que queremos el gráfico en escala logarítmica. Se iguala `log` al eje que se desee tener en escala logarítmica.
- `main("título")`: para poner título al gráfico. Si en vez de un texto queráis poner una expresión matemática, tenéis que utilizar la función `expression()`
- `xlab("etiqueta")`: para poner etiqueta al eje  $X$
- `ylab("etiqueta")`: para poner etiqueta al eje  $Y$
- `pch=n`: significa "point character" y se usa para elegir el símbolo de los puntos.  $n = 0, 1, \dots, 25$ . El valor por defecto es `pch = 1`. En las diapositivas del tema 4 aparecen los símbolos representados.
- `cex`: para elegir el tamaño de los símbolos (puntos o líneas). Ejemplos: `cex=1`, `cex=2` (el doble de grandes)...
- `col="color en inglés"`: para elegir el color del borde de los símbolos. Gama de colores. Ref.: University of Columbia.
- `bg = "color en el interior del punto"`: da el color del interior del punto.

## Ejemplo de escala logarítmica. Modificar una gráfica

```
par(mfrow = c(1,2)) #sirve para colocar dos gráficos en la misma linea
plot = plot(exp(1:20), xlab = "Indice", ylab = expression(e^{1:20}),
            main = "Escala lineal")
plotLog = plot(exp(1:20), log = "y", xlab = "Indice", ylab = expression(e^{1:20}),
              main = "Escala logaritmica en el eje y")
```



```
par(mfrow = c(1,1)) #devuelve los gráficos a como estaban, es obligatorio
#Usa echo="False" para que no aparezca el codigo
```

## Ejemplos varios

### SUCESIÓN DE FIBONACCI

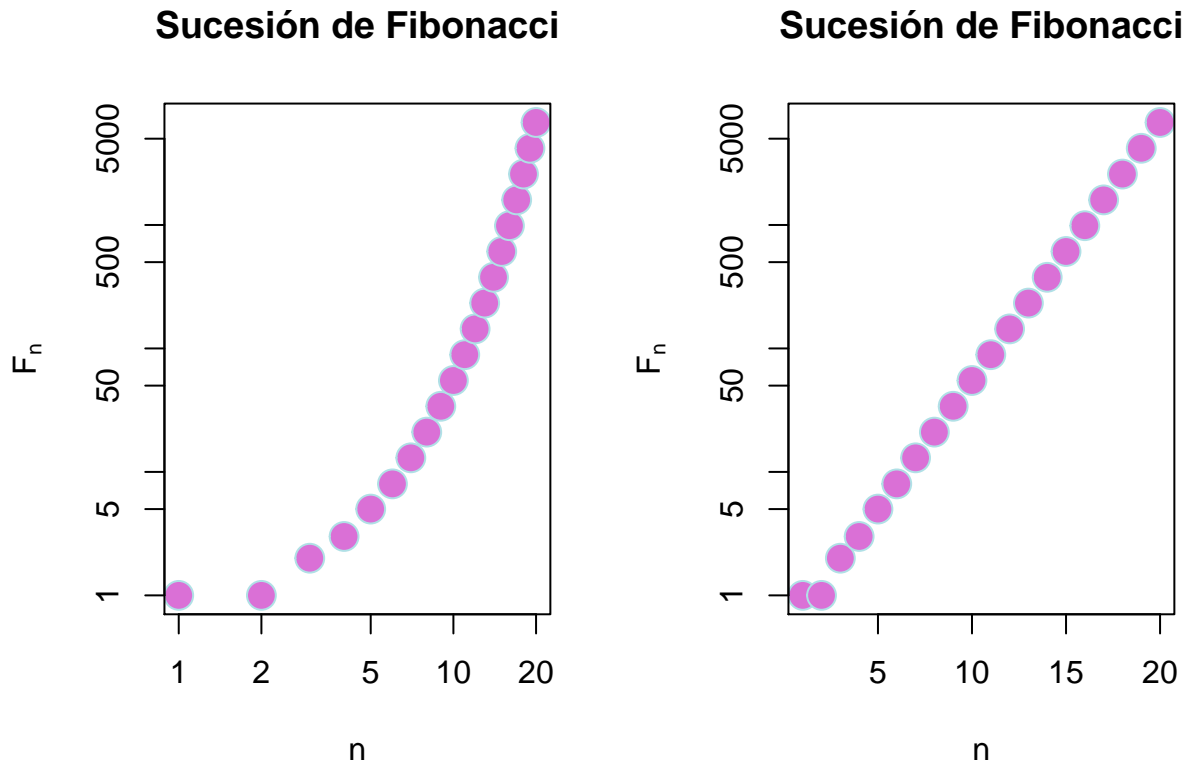
Es una función que se basa en sumar los dos elementos anteriores.

```
n = 1:20
#Expresión sacada de Wikipedia para el cálculo de los n°s fibonacci
fib = (1/sqrt(5))*((1+sqrt(5))/2)^n - (1/sqrt(5))*((1-sqrt(5))/2)^n
#imprimo n°s fibonacci
fib
```

```
## [1] 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
## [16] 987 1597 2584 4181 6765
```

```
par(mfrow = c(1,2))
plot(fib, xlab = "n",
     ylab = expression(F[n]),
     main = "Sucesión de Fibonacci",
     pch = 21, cex = 2,
     col = "powderblue",
     bg = "orchid",
```

```
log = "xy")
plot(fib, xlab = "n",
     ylab = expression(F[n]),
     main = "Sucesión de Fibonacci",
     pch = 21, cex = 2,
     col = "powderblue",
     bg = "orchid", log = "y")
```



```
par(mfrow = c(1,1))
```

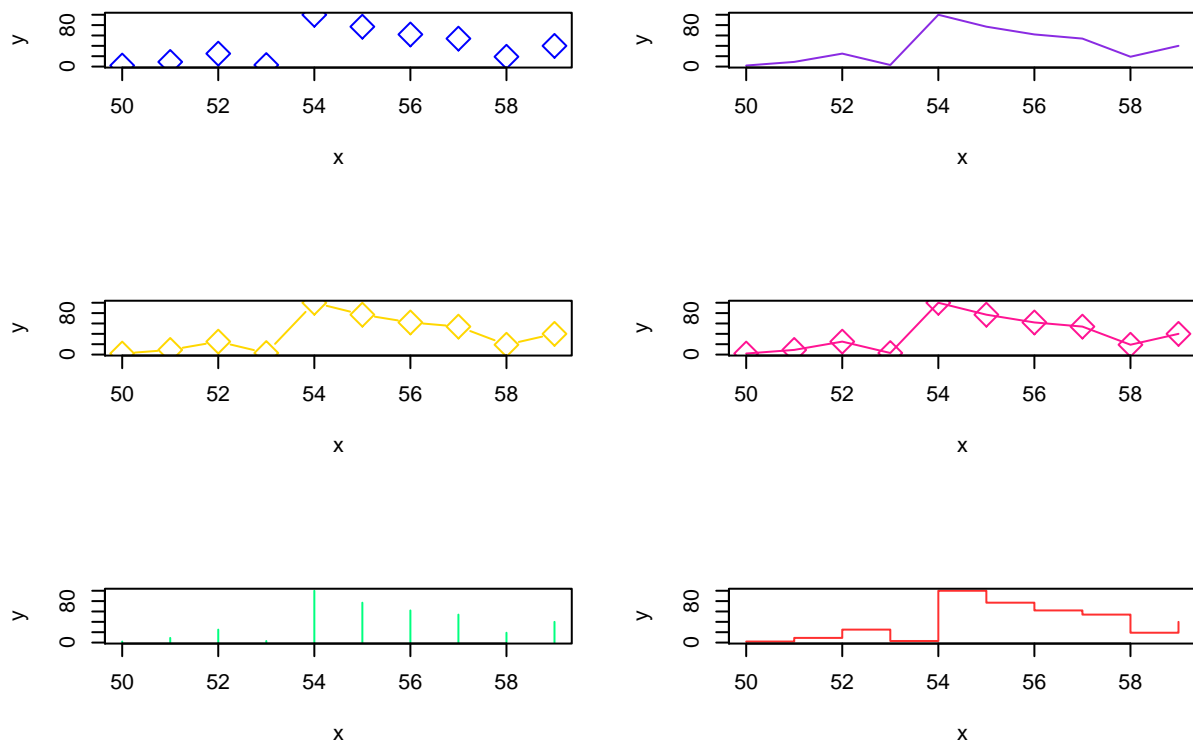
## 2.2 Segunda clase. Tipos de gráficos

Se pueden emplear otros parámetros para seleccionar el tipo de gráfico que queremos representar.

- **type:** para elegir el tipo de gráfico que queremos. Se iguala a estas letras:
  - p: puntos (valor por defecto)
  - l: líneas rectas que unen los puntos (dichos puntos no tienen símbolo)
  - b: líneas rectas que unen los puntos (dichos puntos tienen símbolo). Las líneas no traspasan los puntos
  - o: como el anterior pero en este caso las líneas sí que traspasan los puntos
  - h: histograma de líneas
  - s: histograma de escalones
  - n: para no dibujar los puntos

## Ejemplo de diferentes tipos de gráficos

```
par(mfrow = c(3,2))
x = c(50:59)
y = c(2,9,25,3,100,77,62,54,19,40)
plot(x,y, pch = 23, cex = 2, col = "blue", type = "p")
plot(x,y, pch = 23, cex = 2, col = "blueviolet", type = "l")
plot(x,y, pch = 23, cex = 2, col = "gold", type = "b")
plot(x,y, pch = 23, cex = 2, col = "deeppink", type = "o")
plot(x,y, pch = 23, cex = 2, col = "springgreen", type = "h")
plot(x,y, pch = 23, cex = 2, col = "firebrick1", type = "s")
```



```
par(mfrow = c(1,1))
```

Existen otros parámetros que se aplican al tipo de línea. Sirve por tanto para funciones continuas. Por defecto R te crea 100 puntos seguidos que no son distinguibles y lo vemos como una línea.

- `lty`: para especificar el tipo de línea (l type)
  - “solid” : 1: línea continua (valor por defecto)
  - “dashed” : 2: línea discontinua
  - “dotted” : 3: línea de puntos
  - “dotdashed” : 4: línea que alterna puntos y rayas
- `lwd`: para especificar el grosor de las líneas (l width)

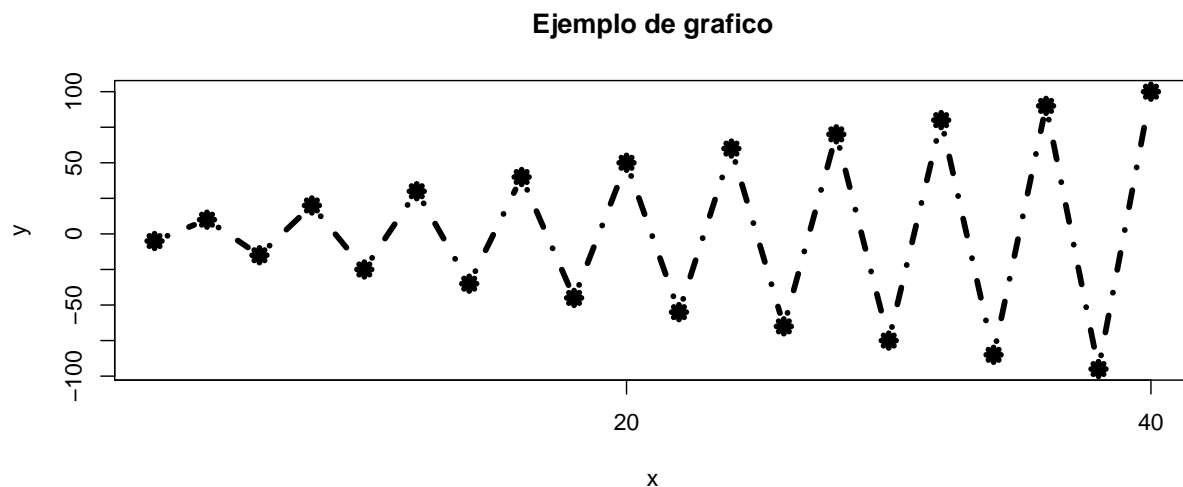


- **xlim:** para modificar el rango del eje *X* (hay que suministrar un vector de 2 puntos para indicar el rango deseado)
- **ylim:** para modificar el rango del eje *Y* (hay que suministrar un vector de 2 puntos para indicar el rango deseado)
- **xaxp:** para modificar posiciones de las marcas en el eje *X* (indicar desde que numero a que numero aparecen y cuantas -> a,b,p)
- **yaxp:** para modificar posiciones de las marcas en el eje *Y* (indicar desde que numero a que numero aparecen y cuantas -> a,b,p)

### Ejemplo ilustrativo

#### Ejemplo tipo de los parámetros de la función plot

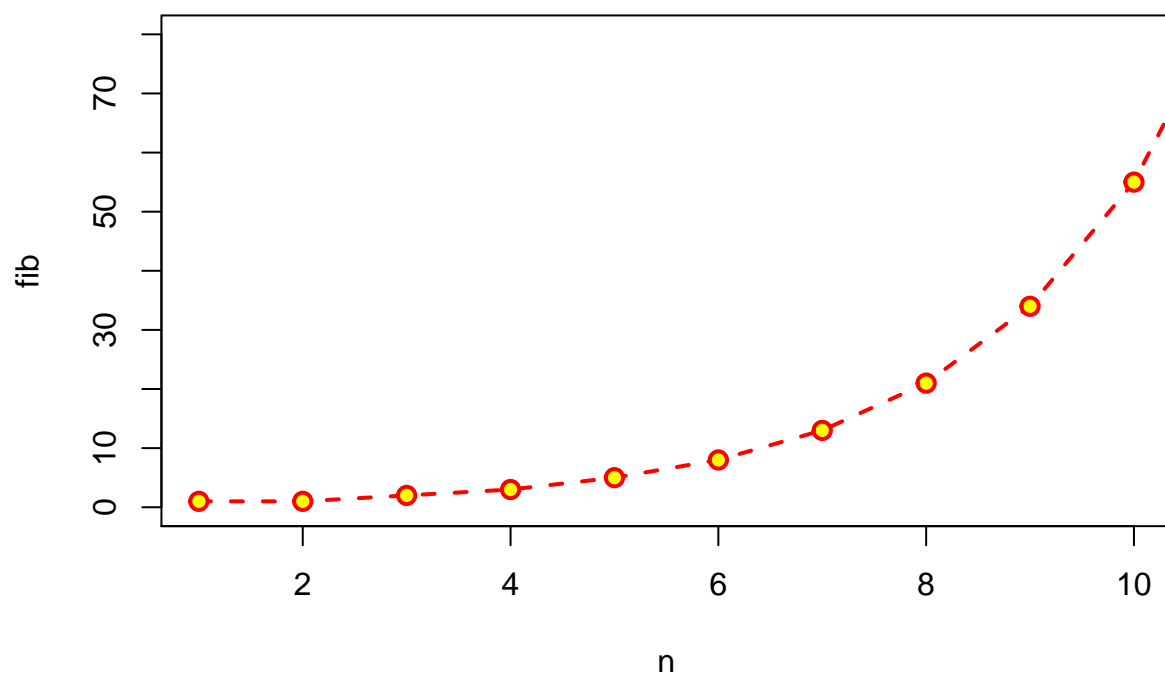
```
x = (2*(1:20))
y = (-1)^(1:20)*5*(1:20)
plot(x,y, main = "Ejemplo de grafico", pch = 8, cex = 1, type = "b", lty = 4, lwd = 4,
     xaxp = c(0,40,2), yaxp = c(-100,100,8))
```



#### Ejemplo con la gráfica de los números Fibonacci

```
plot(n, fib, pch = 21, col = "red",
     bg = "yellow", cex = 1.2,
     main = "Fibonacci",
     type = "o",
     lty = "dashed",
     lwd = 2,
     xlim = c(1,10),
     ylim = c(0, 80),
     xaxp = c(0,10,5),
     yaxp = c(0,80, 8))
```

## Fibonacci



Para consultar los parámetros y determinados aspectos relacionados con la función `plot` empleando los comandos en R de `?plot` y `?par`.