

Introducción a la representación gráfica en R

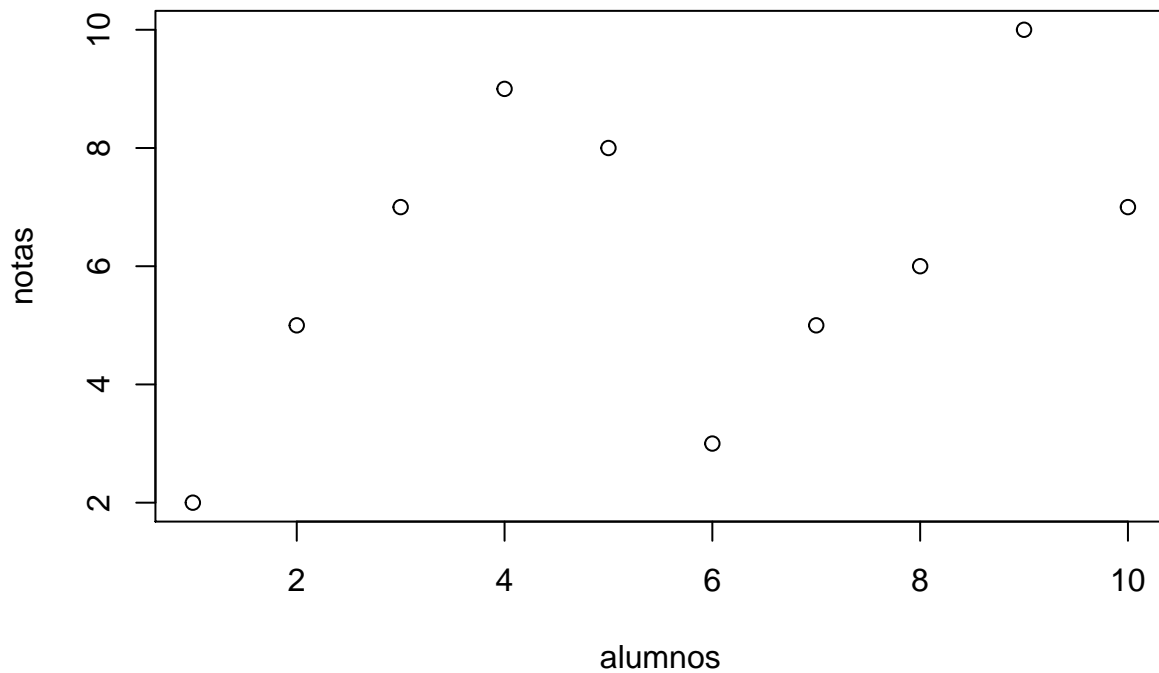
Jordi Vanrell

25/10/2020

La función plot

- `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

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



Para colocar el pie de foto:

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

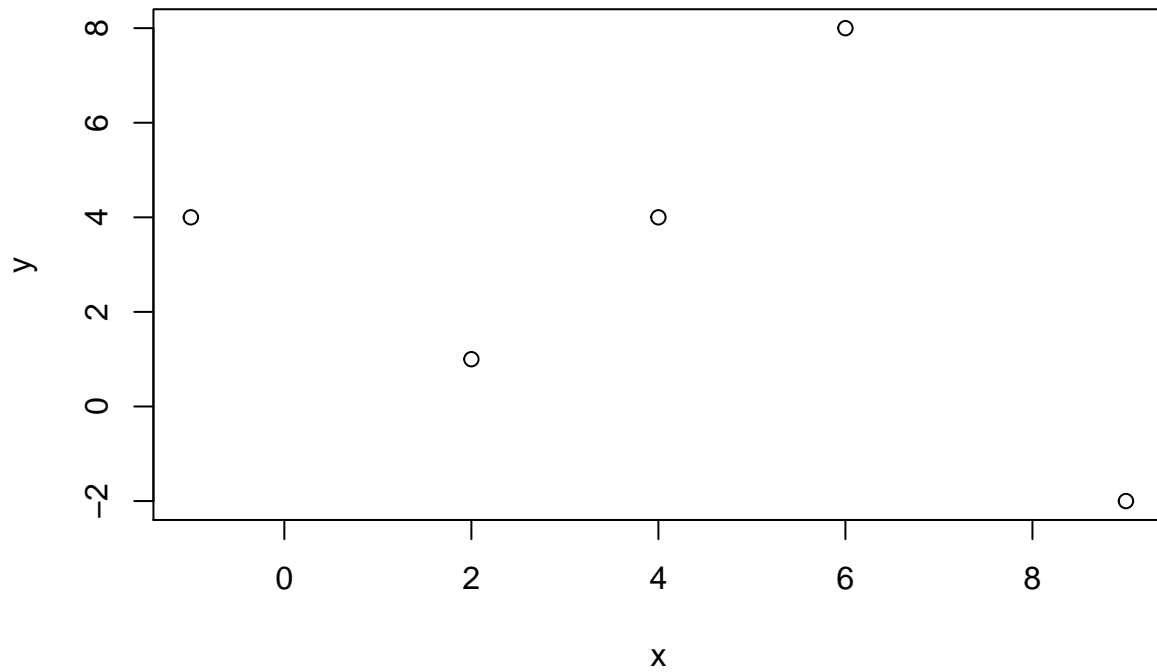
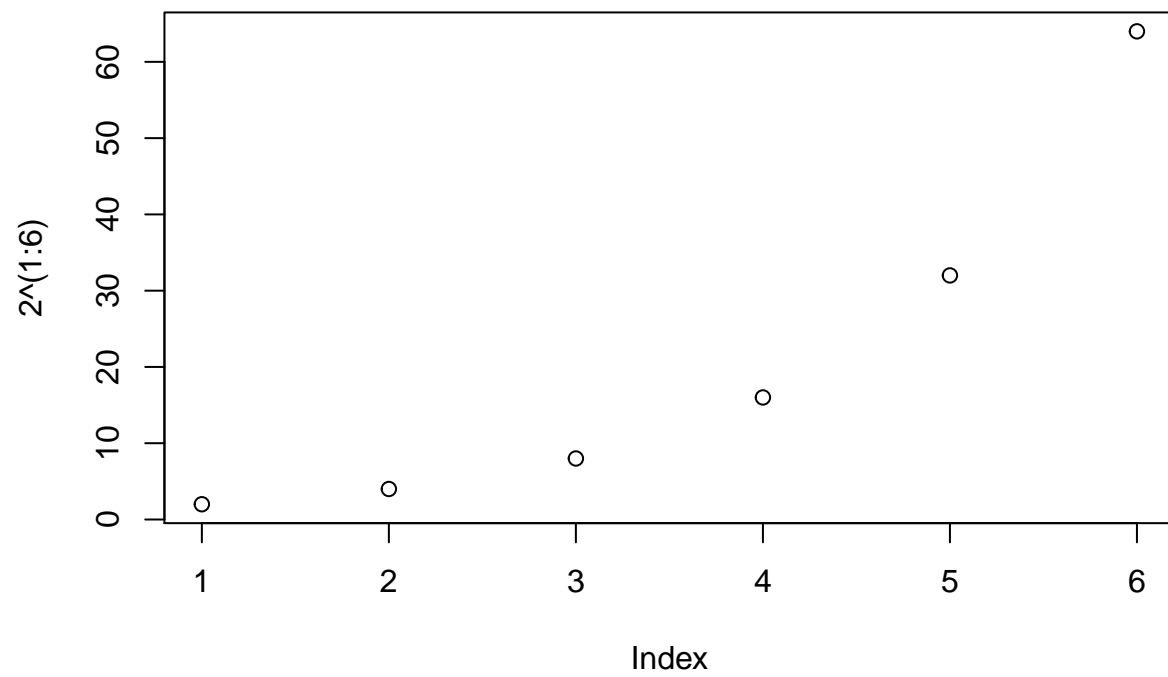


Figure 1: Gráfico básico explicando el uso de plot

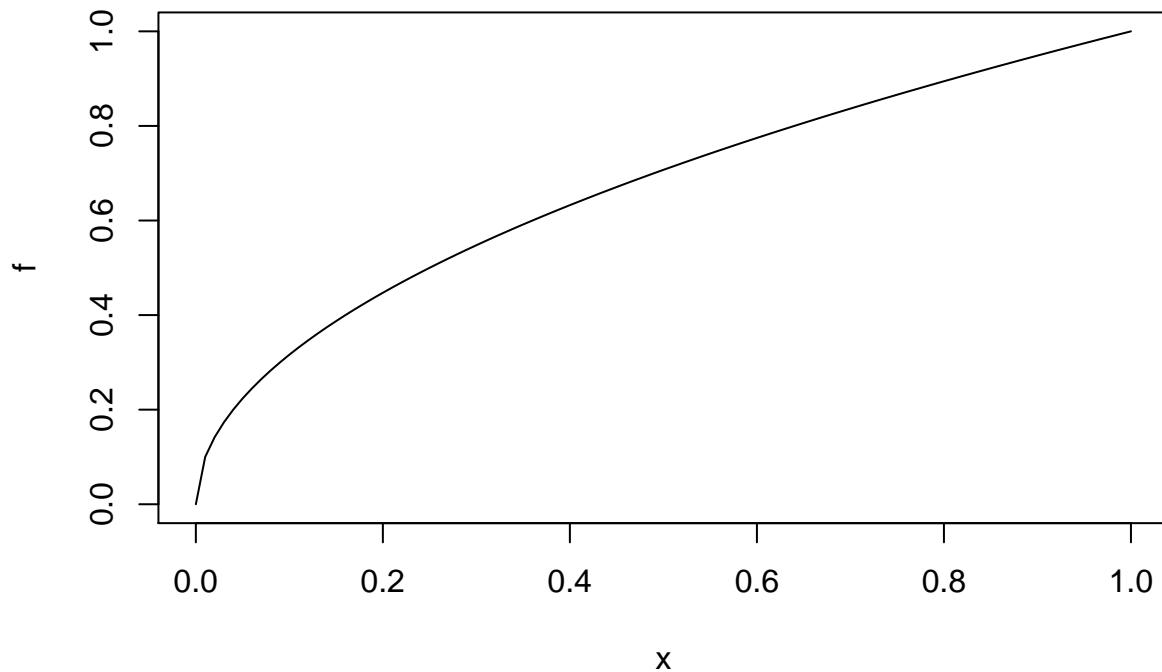
Si no incorporamos vector y, Rnos tomará x como si fuera el vector de datos y.

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



Si se quiere hacer plot de una función $f(x)$:

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

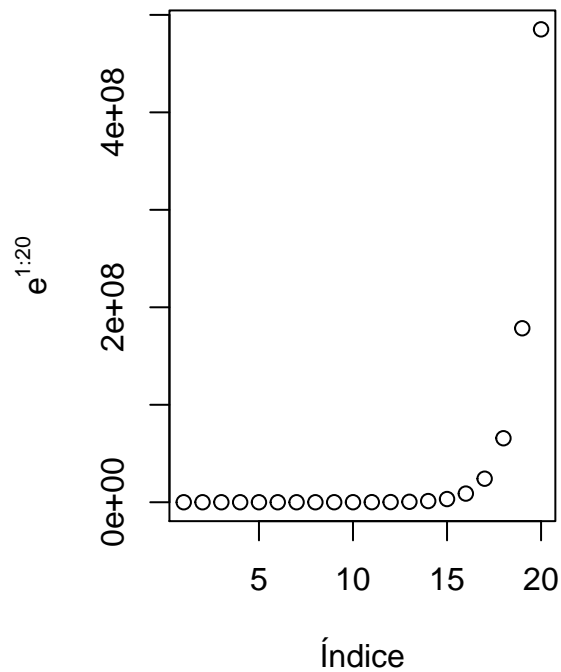


Parámetros de la función plot

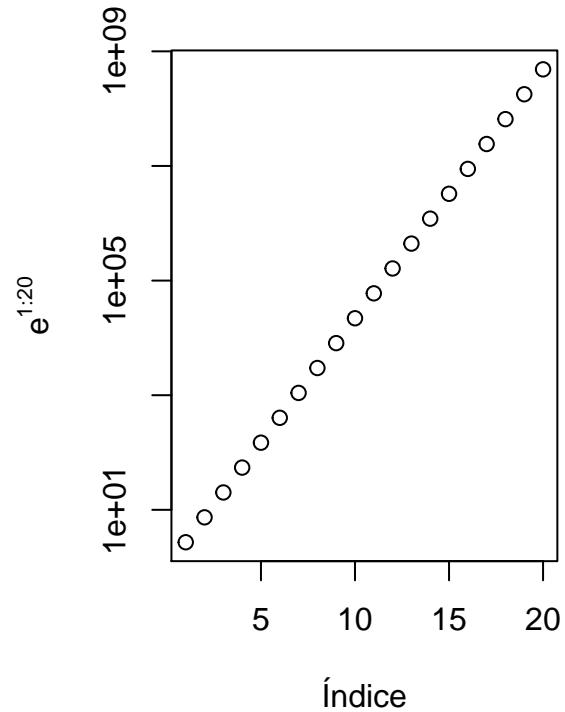
- `log`: para indicar que queremos el gráfico en escala logarítmica.
- `main("título")`: para poner título al gráfico. Si en vez de un texto queráis poner una expresión matemática, tenéis que utilizar la función `expression()`, que puede ser en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- `xlab("etiqueta")`: para poner etiqueta al eje X
- `ylab("etiqueta")`: para poner etiqueta al eje Y
- `pch=n`: para elegir el símbolo de los puntos. $n=0,1,\dots,25$. El valor por defecto es `pch = 1`
- `cex`: para elegir el tamaño de los símbolos
- `col="color en inglés"`: para elegir el color de los símbolos.

```
par(mfrow=c(1,2)) #esto sirve para representar un gráfico al lado del otro (una fila y dos columnas)
plot=plot(exp(1:20),xlab="Índice",ylab=expression(e^{1:20}),main="Escala lineal")
plotLog=plot(exp(1:20),log="y",xlab="Índice",ylab=expression(e^{1:20}),main="Escala logarítmica en el e
```

Escala lineal



Escala logarítmica en el eje y



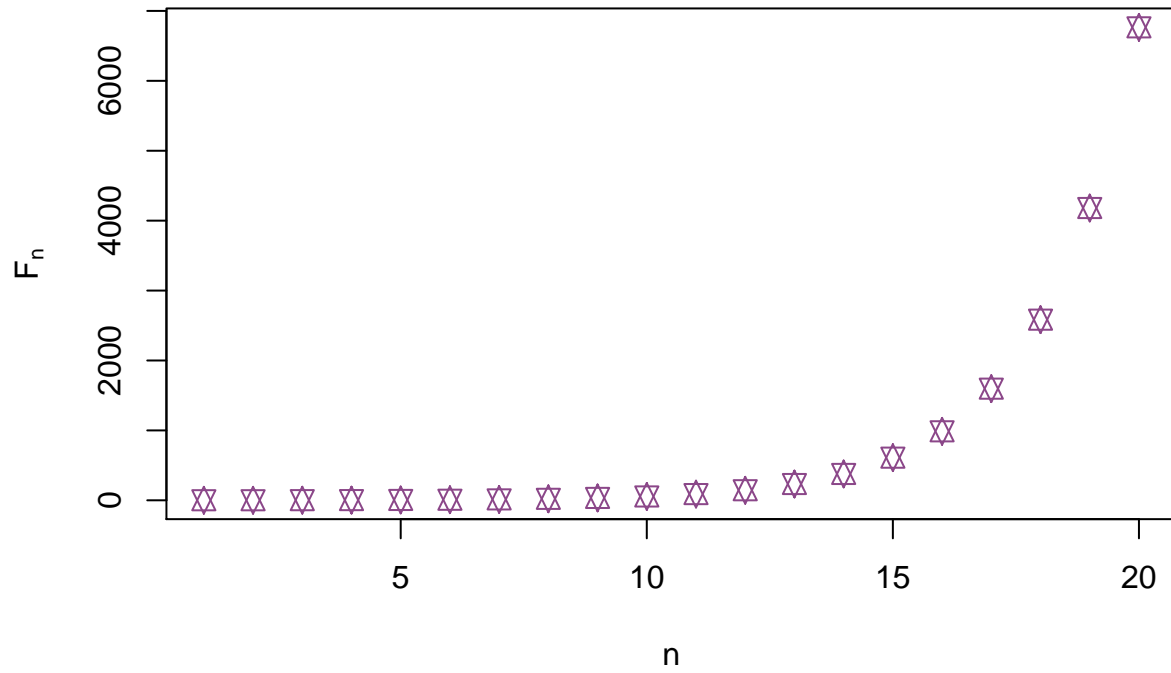
```
par(mfrow=c(1,1)) #Para anular la representación de gráficos anterior.
```

```
n=1:20
fib=(1/sqrt(5))*((1+sqrt(5))/2)^n - (1/sqrt(5))*((1-sqrt(5))/2)^n
fib
```

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

```
plot(fib,xlab="n",ylab=expression(F[n]),main="Sucesión de Fibonacci",pch=11,cex=1.2,col="orchid4")
```

Sucesión de Fibonacci

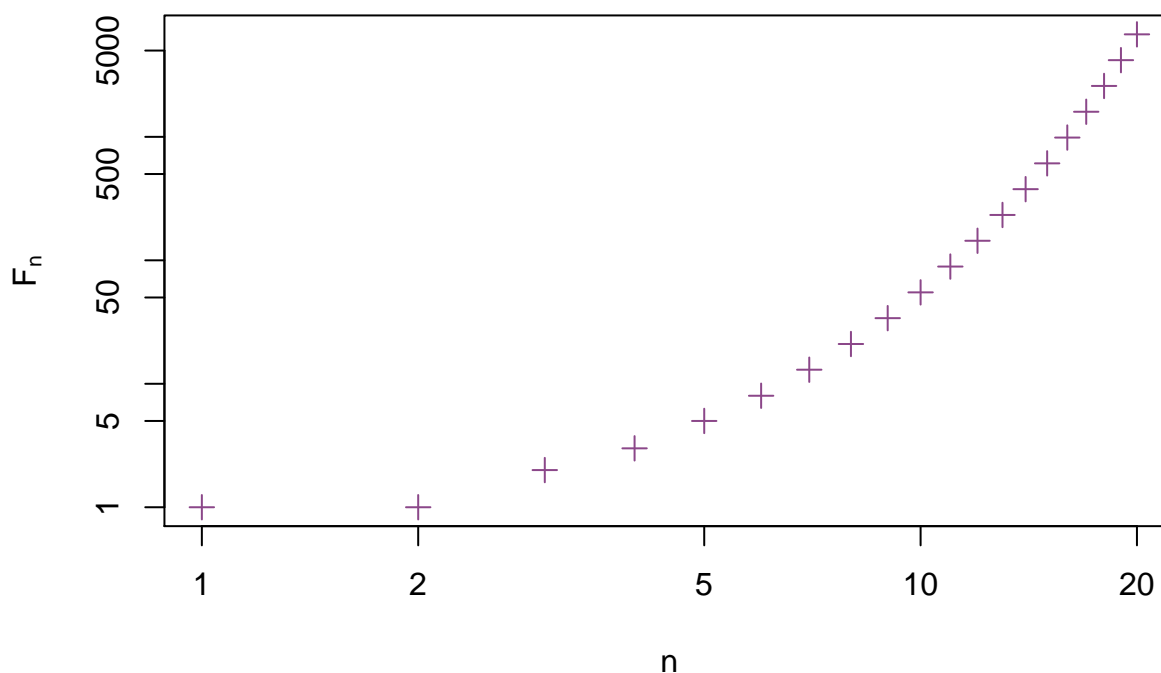


```
n=1:20
fib=(1/sqrt(5))*((1+sqrt(5))/2)^n - (1/sqrt(5))*((1-sqrt(5))/2)^n
fib
```

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

```
plot(fib,xlab="n",ylab=expression(F[n]),main="Sucesión de Fibonacci",pch=3,cex=1.2,col="orchid4", log="n")
```

Sucesión de Fibonacci



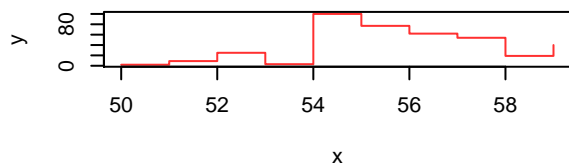
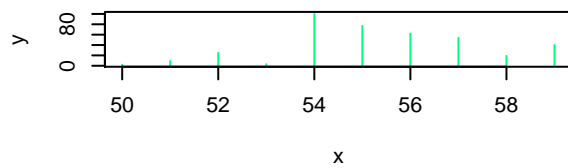
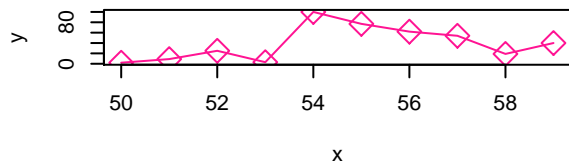
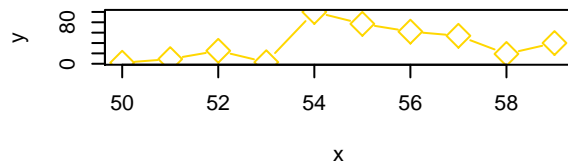
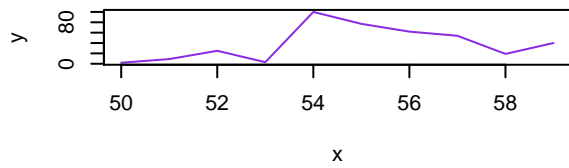
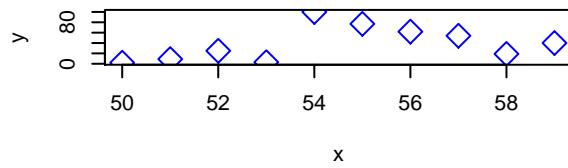
#log=xy introduce logaritmos en ambos ejes.

Más parámetros de representación gráfica

type: para elegir el tipo de gráfico que queremos:

- 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

```
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))
```

lty: para especificar el tipo de línea

- "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

xlim: para modificar el rango del eje X

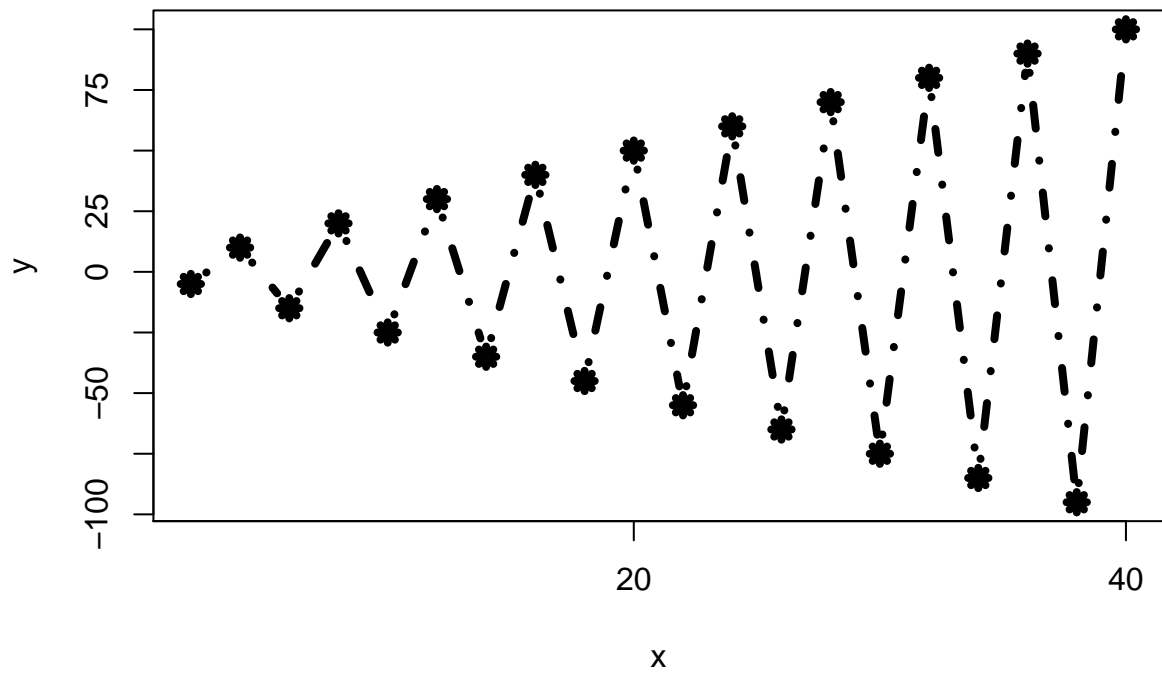
ylim: para modificar el rango del eje Y

xaxp: para modificar posiciones de las marcas en el eje X

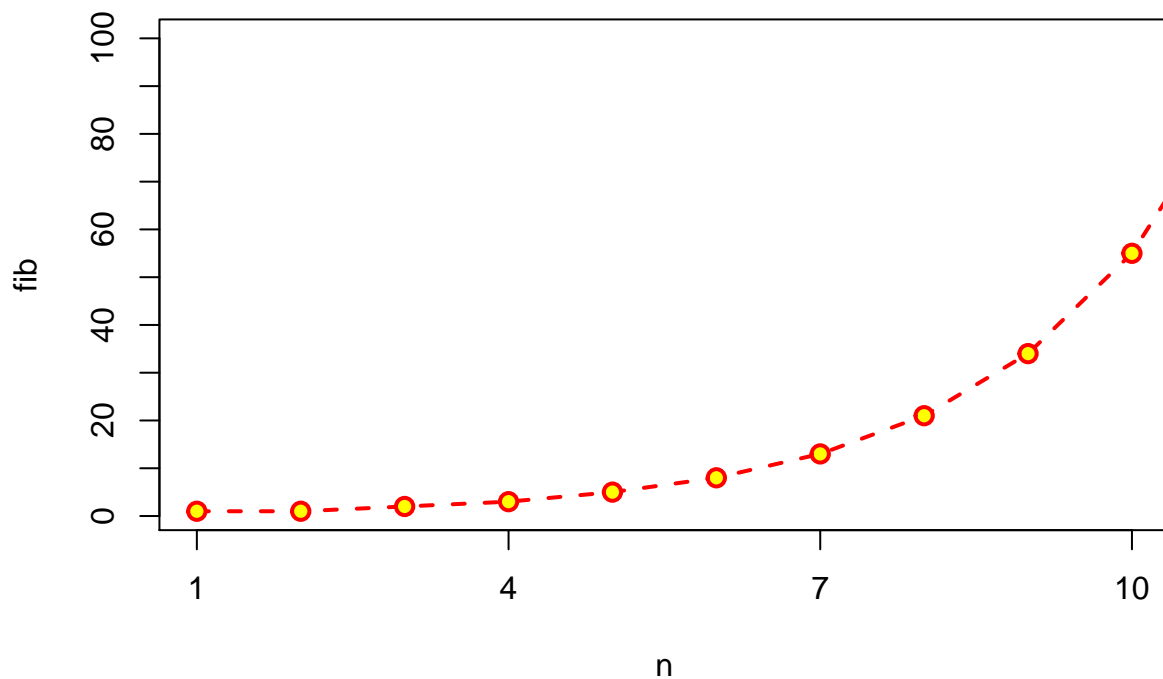
yaxp: para modificar posiciones de las marcas en el eje Y

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


Ejemplo de gráfico



```
plot(n,fib,pch=21,col="red",bg="yellow",cex=1.2,  
     type="o",lty="dashed",lwd=2,  
     xlim=c(1,10),ylim=c(1,100),  
     xaxp=c(1,10,3),yaxp=c(0,100,10))
```

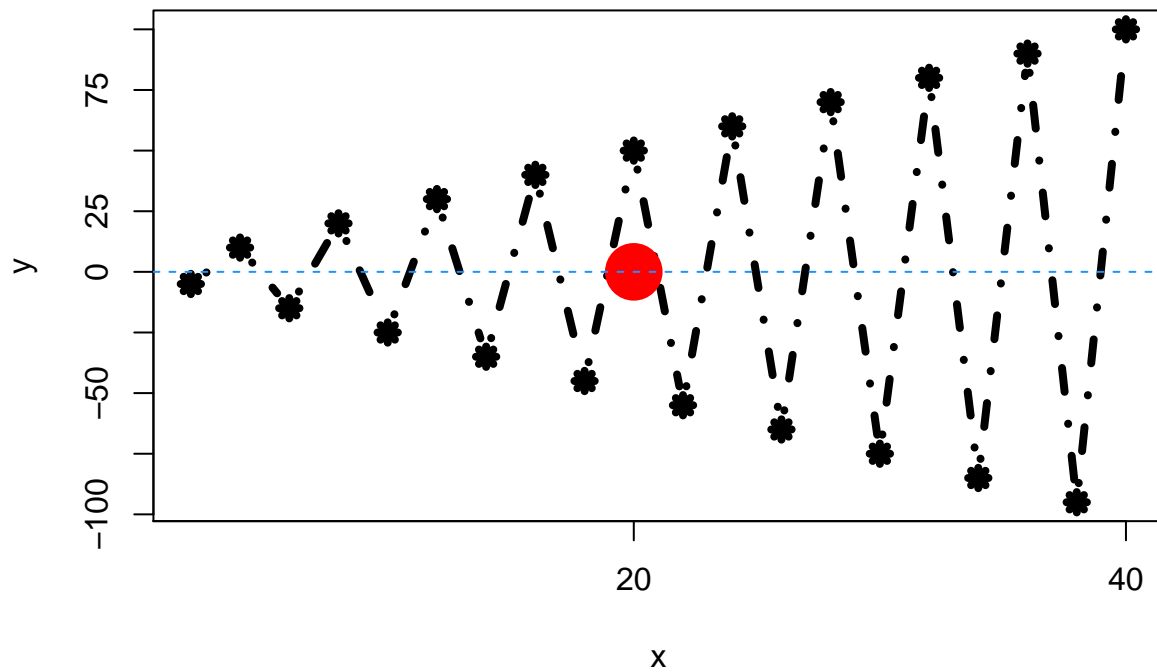


Añadir elementos a un gráfico

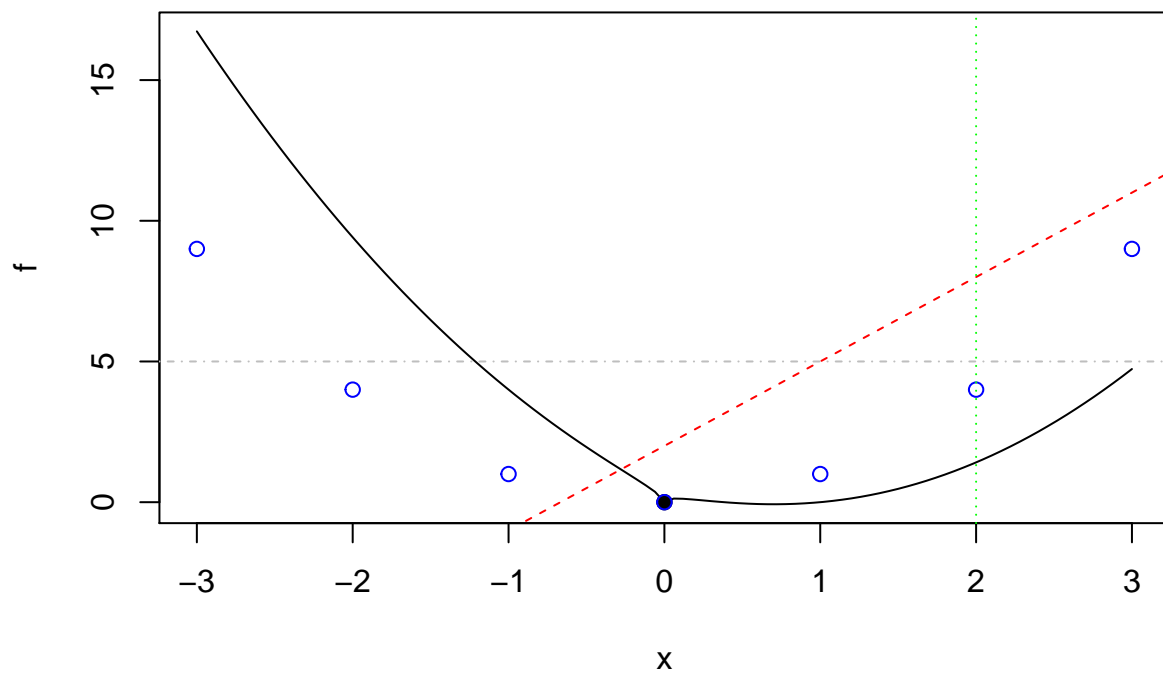
- `points(x,y)`: añade un punto de coordenadas (x,y) a un gráfico ya existente
- `abline`: para añadir una recta a un gráfico ya existente
- `abline(a,b)`: añade la recta $y=ax+b$
- `abline(v = x0)`: añade la recta vertical $x=x0$. `v` puede estar asignado a un vector.
- `abline(h = y0)`: añade la recta horizontal $y=y0$. `h` puede estar asignado a un vector

```
x=(2*(1:20))
y=(-1)^(1:20)*5*(1:20)
plot(x,y,main="Ejemplo de gráfico",pch=8,cex=1,type="b",lty=4,lwd=4,xaxp=c(0,40,2),yaxp=c(-100,100,8))
points(20,0,col="red",cex=4,pch=16)
abline(h=0,lty=2,col="dodgerblue")
```

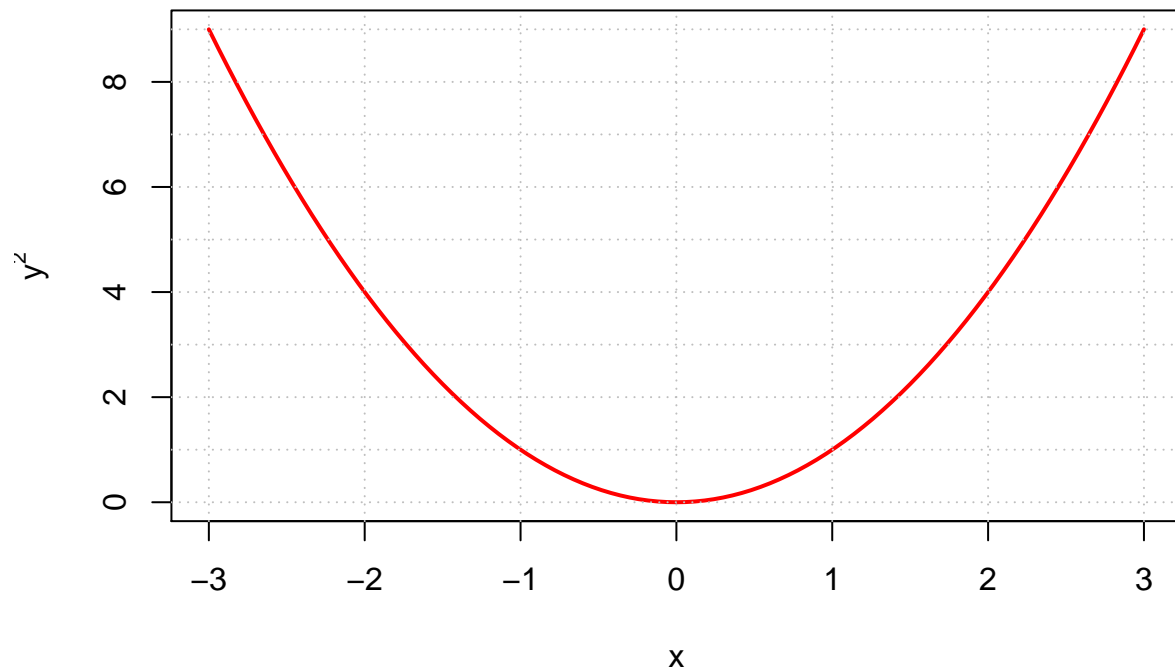
Ejemplo de gráfico



```
f<-function(x){x^2-2*x+sqrt(abs(x))}
plot(f,xlim=c(-3,3))
points(0,0,pch=19)
points(-3:3,(-3:3)^2,col="blue")
abline(2,3,lty="dashed",col="red")
abline(v=2,lty="dotted",col="green")
abline(h=5,lty="dotdash",col="grey")
```



```
f<-function(x){x^2}
plot(f,xlim=c(-3,3),col="red",lwd=2,ylab=expression(y^2),xlab="x")
abline(h=0:9,v=-3:3,lty="dotted",col="grey")
```



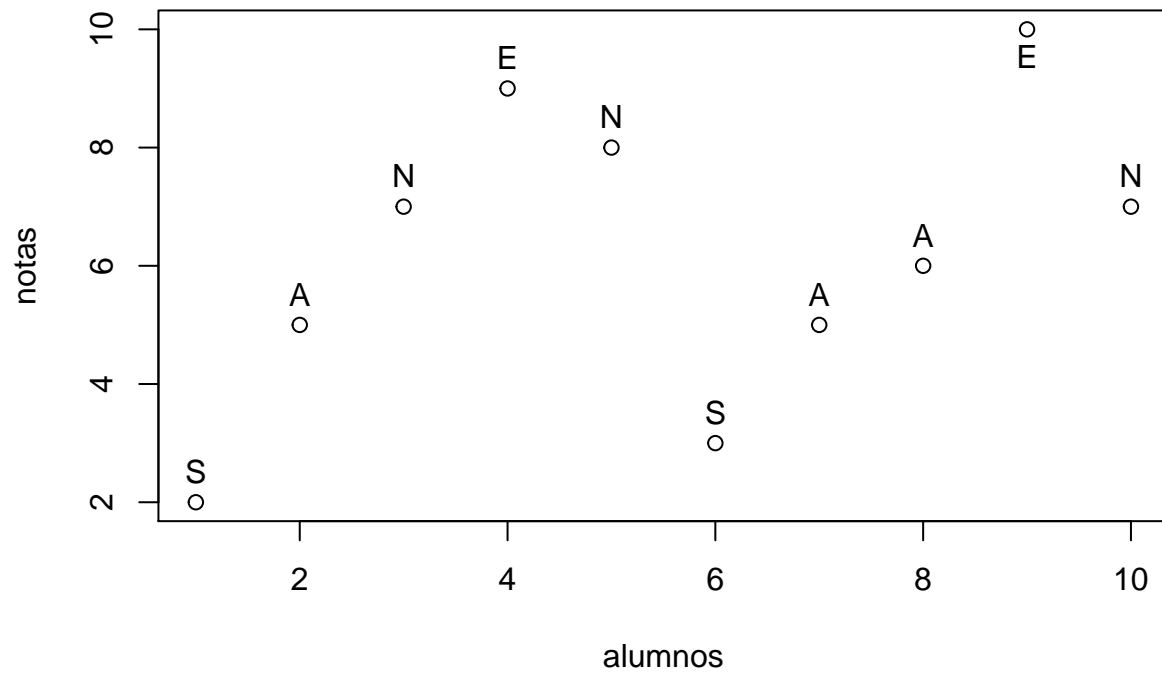
`text(x,y,labels = "...")`: añade en el punto de coordenadas (x,y) el texto especificado como argumento de labels

- `pos`: permite indicar la posición del texto alrededor de las coordenadas (x,y). Admite los siguientes valores:

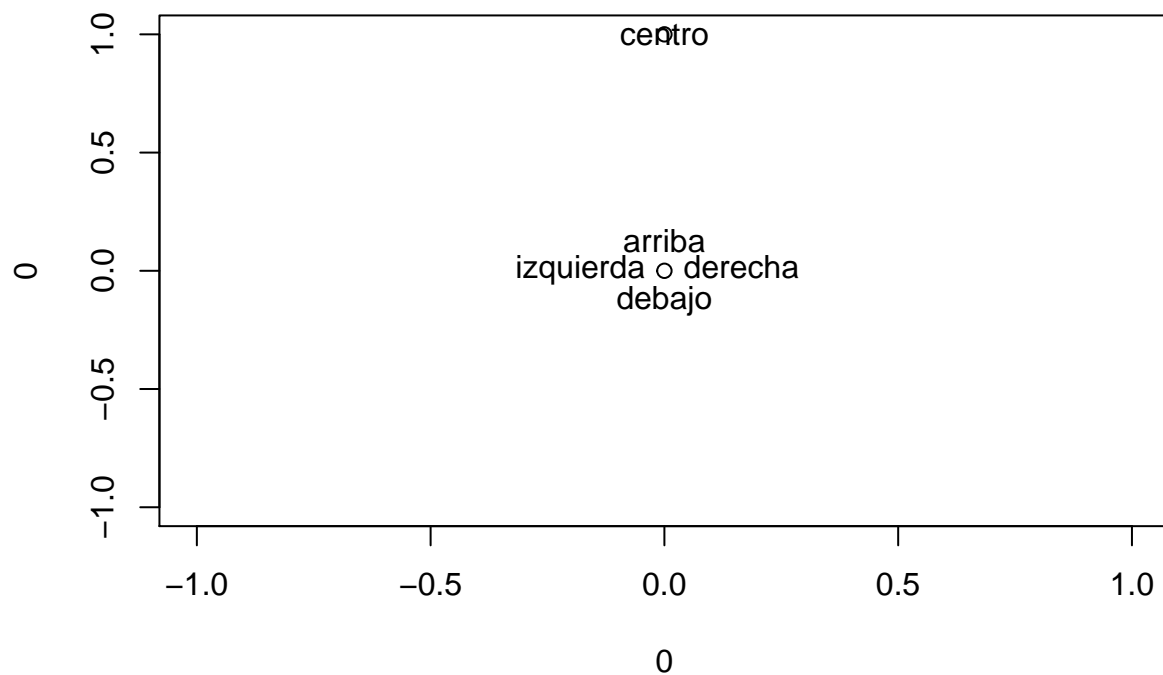
- 1: abajo
- 2: izquierda
- 3: arriba
- 4: derecha
- 5: sin especificar: el texto se sitúa centrado en el punto (x,y)

```
alumnos = c(1:10)
notas = c(2,5,7,9,8,3,5,6,10,7)
plot(alumnos,notas, main = "Grafico con texto")
text(alumnos,notas, labels = c("S","A","N","E","N","S","A","A","E","N"),
     pos = c(rep(3,times = 8),1,3))
```

Grafico con texto



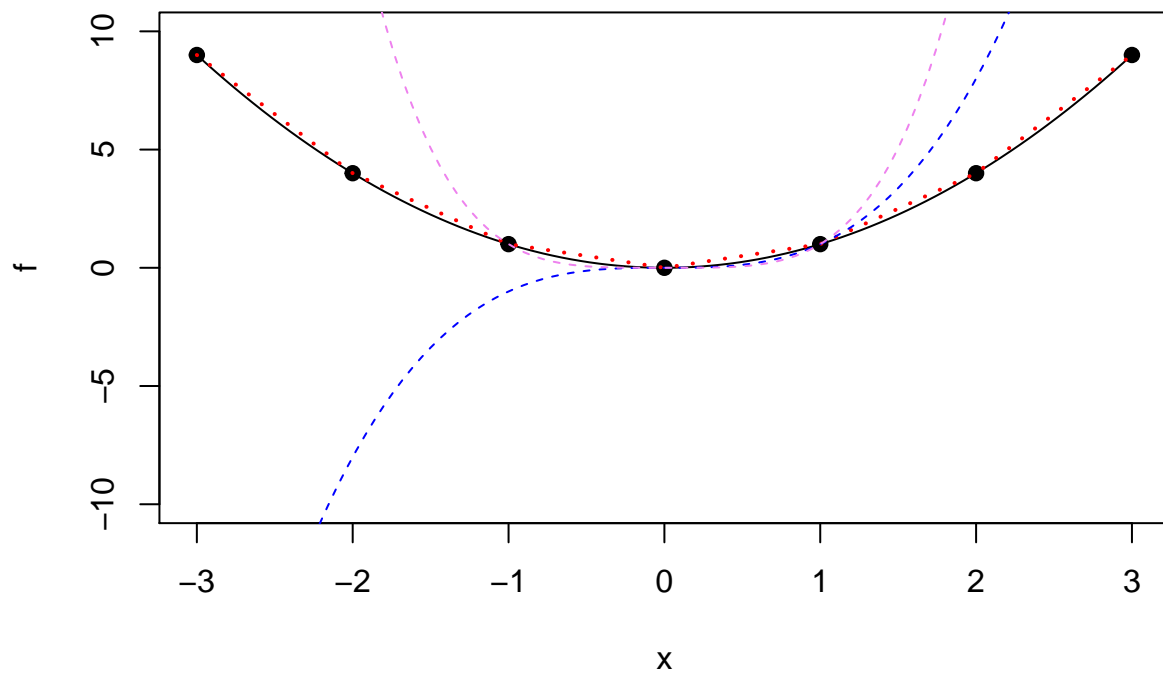
```
plot(0,0)
text(0,0,labels="debajo",pos=1)
text(0,0,labels="izquierda",pos=2)
text(0,0,labels="arriba",pos=3)
text(0,0,labels="derecha",pos=4)
points(0,1)
text(0,1,labels="centro")
```



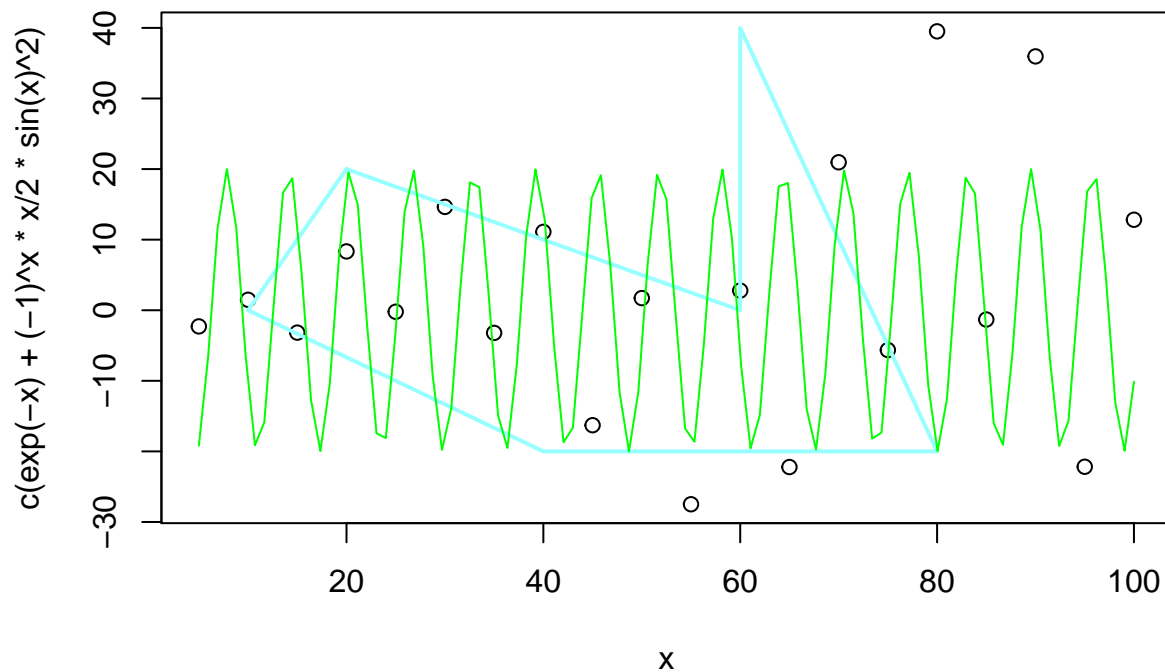
`lines(x, y)`: añade a un gráfico existente una **línea poligonal** que une los puntos (x_i, y_i) sucesivos. x, y son vectores numéricos

`curve(curva)`: permite añadir la gráfica de una curva a un gráfico existente. Debe especificarse `add=TRUE`. Si no, la curva no se añade. La curva se puede especificar mediante una expresión algebraica con variable x , o mediante su nombre si la hemos definido antes.

```
f<-function(x){x^2}
plot(f,xlim=c(-3,3),ylim=c(-10,10))
points(-3:3,f(-3:3),pch=19)
lines(-3:3,f(-3:3),lwd=2,lty="dotted",col="red")
curve(x^3,lty="dashed",col="blue",add=TRUE)
curve(x^4,lty="dashed",col="violet",add=TRUE)
```



```
x = c(5*(1:20))
plot(x,c(exp(-x)+(-1)^x*x/2*sin(x)^2))
lines(c(20,10,40,80,60,60,20),c(20,0,-20,-20,40,0,20), lwd = 2, col = "darkslategray1")
curve(20*sin(x), add = TRUE, col = "green")
```

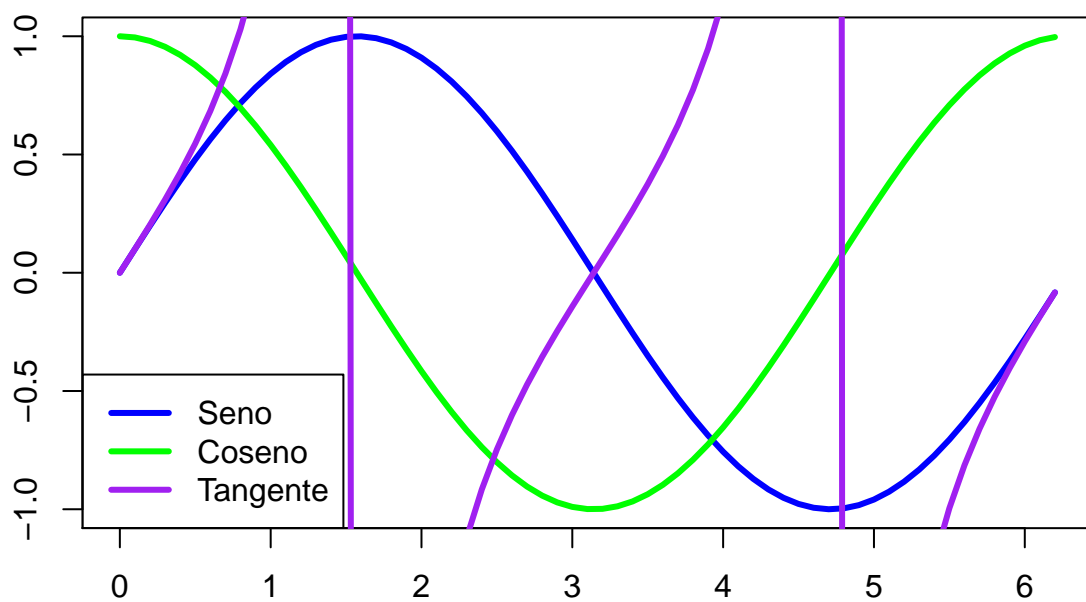



`legend(posición, legend = ...)`: para añadir una leyenda.

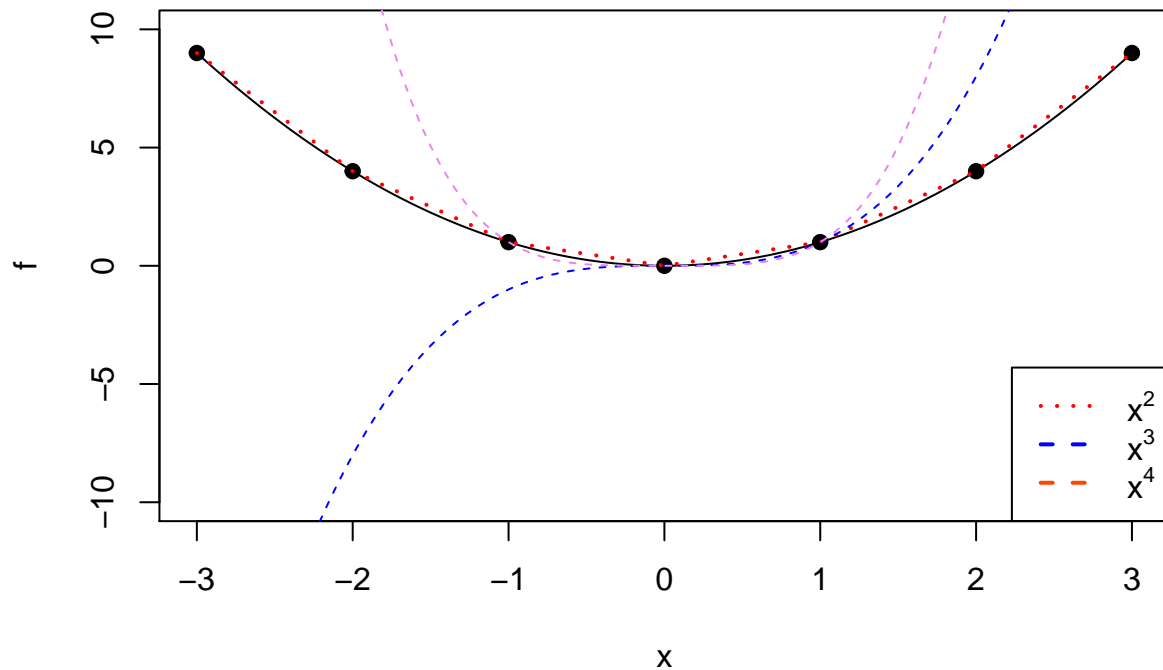
La posición indica donde queremos situar la leyenda. Puede ser o bien las coordenadas de la esquina superior izquierda de nuestra leyenda, o bien una de las palabras siguientes: “bottom” / “bottomright” / “bottomleft” / “top” / “topright” / “topleft” / “center” / “right” / “left”.

`legend`: contiene el vector de nombres entre comillas con los que queremos identificar a las curvas en la leyenda.

```
x = seq(0,2*pi,0.1)
plot(x,sin(x),type="l",col="blue",lwd=3, xlab="", ylab="")
lines(x,cos(x),col="green",lwd=3)
lines(x, tan(x), col="purple",lwd=3)
legend("bottomleft",col=c("blue","green","purple"), legend=c("Seno","Coseno", "Tangente"),
      lwd=3, bty="l")
```



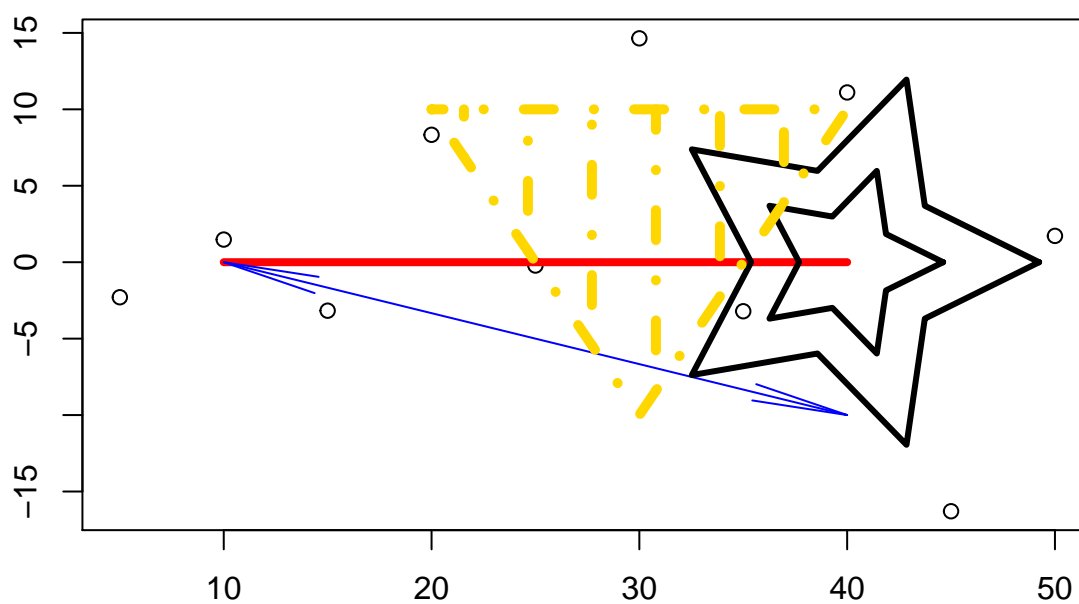
```
f<-function(x){x^2}
plot(f,xlim=c(-3,3),ylim=c(-10,10))
points(-3:3,f(-3:3),pch=19)
lines(-3:3,f(-3:3),lwd=2,lty="dotted",col="red")
curve(x^3,lty="dashed",col="blue",add=TRUE)
curve(x^4,lty="dashed",col="violet",add=TRUE)
legend("bottomright",legend=c(expression(x^2),expression(x^3),expression(x^4)),col=c("red","blue","orange"))
```



- `segments`: para añadir segmentos a un gráfico existente.
- `arrows`: para añadir flechas a un gráfico existente.
- `symbols`: para añadir símbolos a un gráfico existente.
- `polygon`: para añadir polígonos cerrados especificando sus vértices a un gráfico existente.

```
x = c(5*(1:10))
plot(x,c(exp(-x)+(-1)^x*x/2*sin(x)^2), xlab = "", ylab = "",
      main = "Grafico con varios elementos")
segments(10,0,40,0, col = "red", lwd = 4)
arrows(10,0,40,-10, col = "blue", length = 0.5, angle = 5, code = 3)
symbols(40,0,stars = cbind(1,.5,1,.5,1,.5,1,.5,1,.5), add = TRUE, lwd = 3, inches = 0.5)
symbols(40,0,stars = cbind(1,.5,1,.5,1,.5,1,.5,1,.5), add = TRUE, lwd = 3)
polygon(c(20,30,40),c(10,-10,10), col = "gold", density = 3, angle = 90, lty = 4,
        lwd = 5)
```

Grafico con varios elementos



```
alumnos = c(1:10)
notas = c(2,5,7,9,8,3,5,6,10,7)
plot(alumnos,notas, main = "Grafico con texto")
text(alumnos,notas, labels = c("S","A","N","E","N","S","A","A","E","N"),
      pos = c(rep(3,times = 8),1,3))
arrows(9,8,9,6, col = "red", length = 0.1, angle = 45, code = 1)
```

Grafico con texto

