

Graficos

Adrian Vitys

13/1/2022

Gráficos con la función plot

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

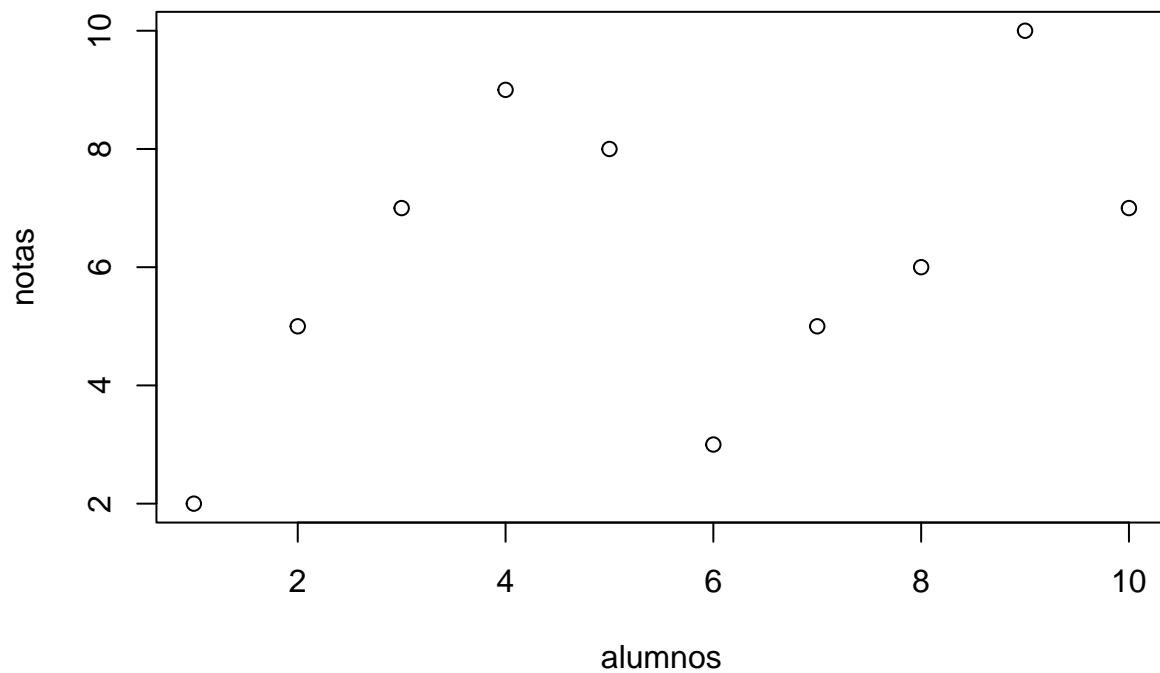
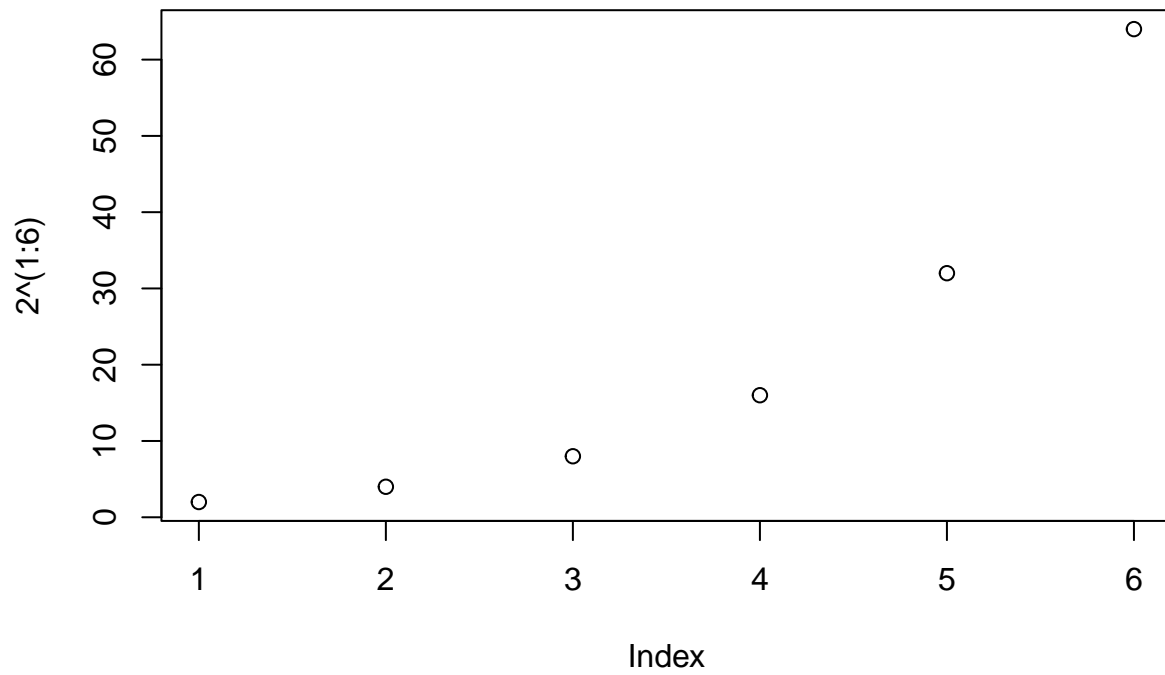


Figure 1: Gráfico básico sobre notas y alumnos

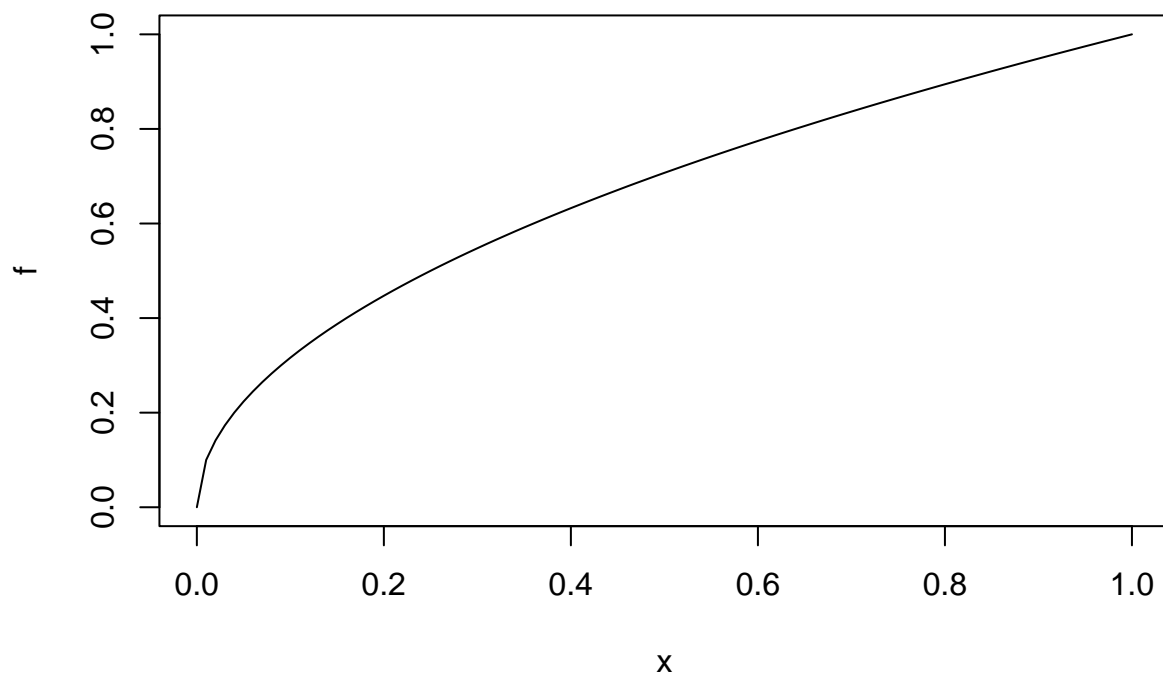
Si no incorporamos vector `y`, `R` nos va a tomar el parámetro `x` como si fuese el vector de datos `y` : `plot(1:n, x)`

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



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

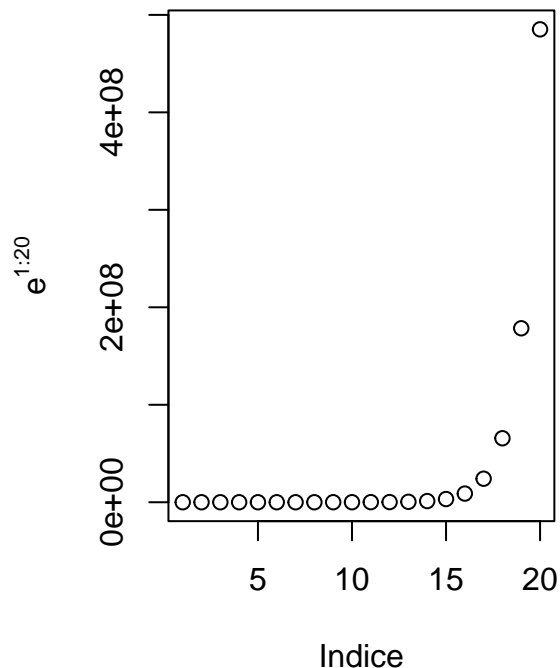
```
f <- function(x){sqrt(x)}  
plot(f)
```



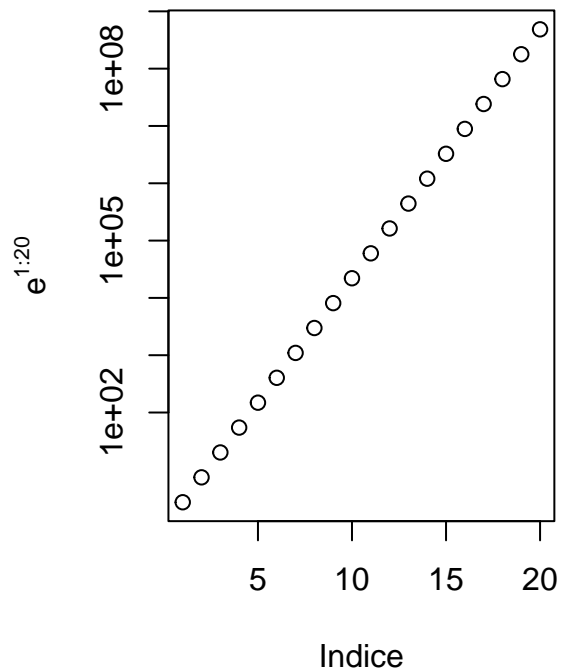
Escala logaritmica

```
# Para hacer los graficos en 1x2
par(mfrow=c(1,2))
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")
```

Escala lineal



Escala logarítmica en el eje y



Parámetros de plot

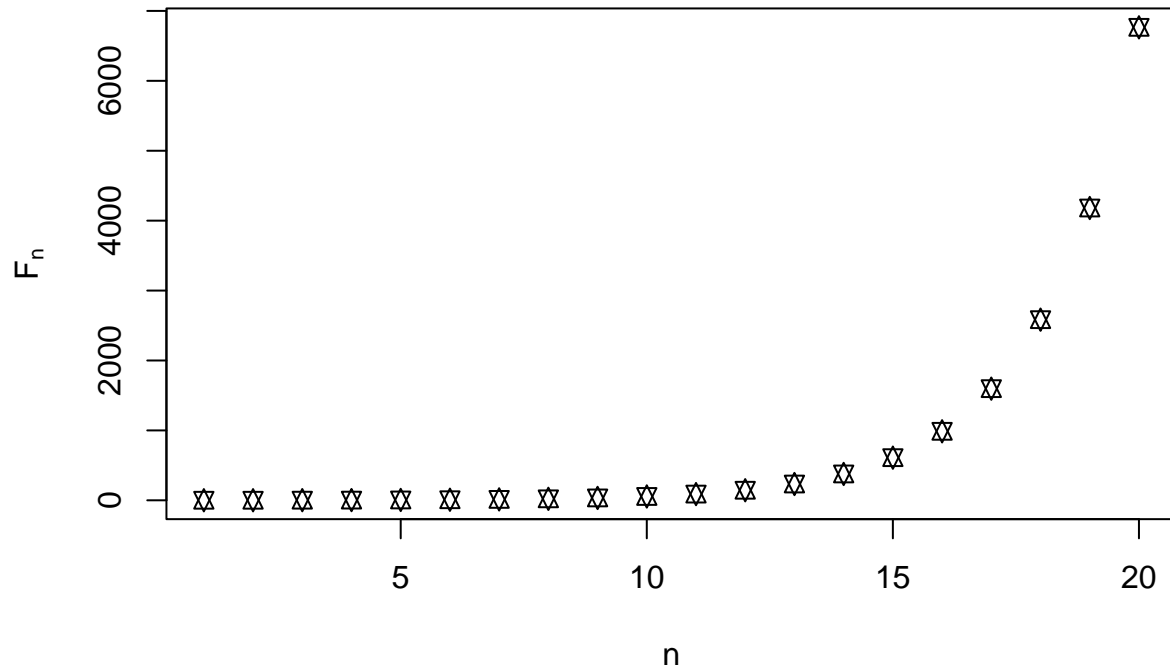
Los parámetros de plot son: log -> Gráfico en escala logarítmica. main("Titulo") -> Poner titulo texto al grafico. expression() -> Poner titulo expresion matematica al grafico. xlab("etiqueta") -> Etiqueta eje X ylab("etiqueta") -> Etiqueta eje Y pch=n -> Simbolo de los puntos. n = 0,1,...,25. Defecto n = 1. cex -> Tamaño del simbolo col="Color en ingles" -> Color de los simbolos type = -> Elegir el tipo de grafico - p = puntos (Default) - l = lineas rectas que unen los puntos (puntos sin simbolo) - b = lineas rectas que unen los puntos (puntos con simbolo) - o = lineas rectas que unen los puntos (puntos con simbolo, lineas traspasan los simbolos) - h = histograma de lineas - s = histograma de escalones - n = no dibujar los puntos lty -> especificar el tipo de linea - "solid":1: linea continua (Default) - "dashed":2: linea discontinua - "dotted":3: linea de puntos - "dotted":4: linea alterna puntos y rayas lwd -> grosor de las lineas xlim -> modificar el rango del eje X ylim -> modificar el rango del eje Y xaxp -> modificar posiciones de las marcas del eje X yaxp -> modificar posiciones de las marcas del eje Y

```
# 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
```

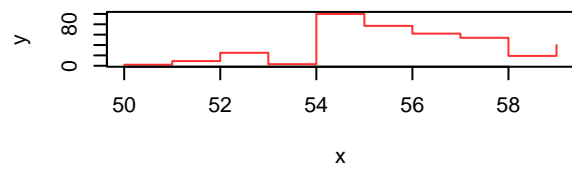
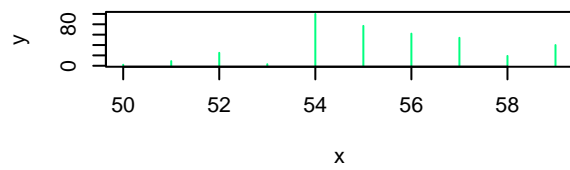
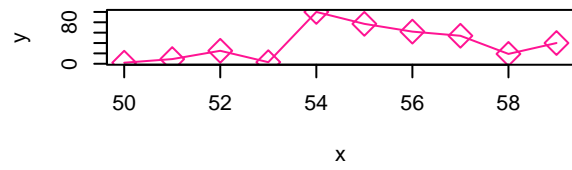
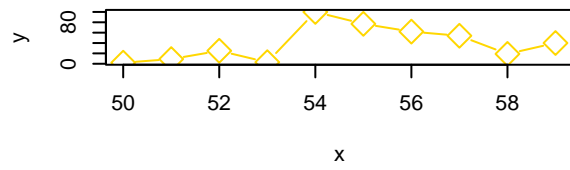
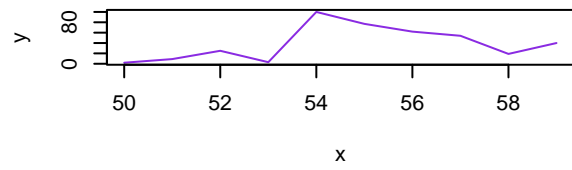
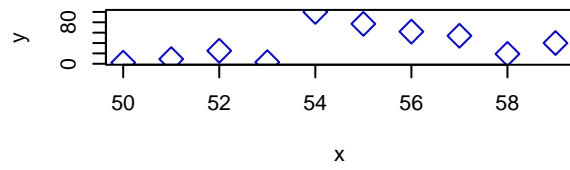
```
plot(fib, xlab = "n", ylab = expression(F[n]),
     main = "Sucesión de Fibonacci", pch = 11)
```

Sucesión de Fibonacci



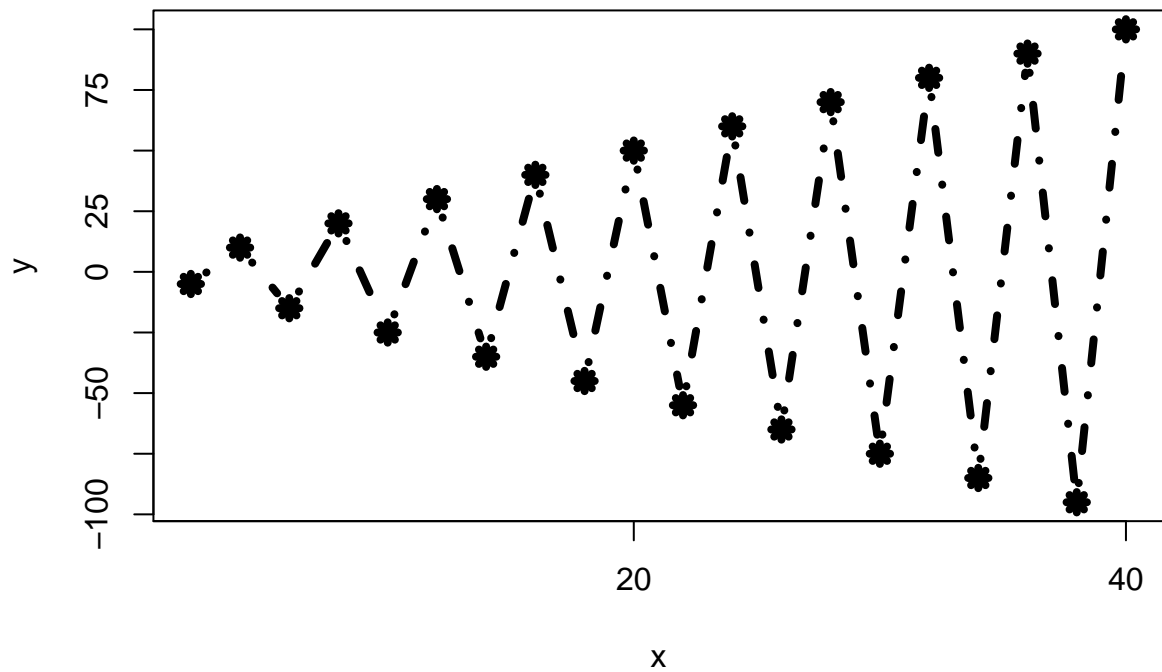
Tipos de Graficos

```
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")
```



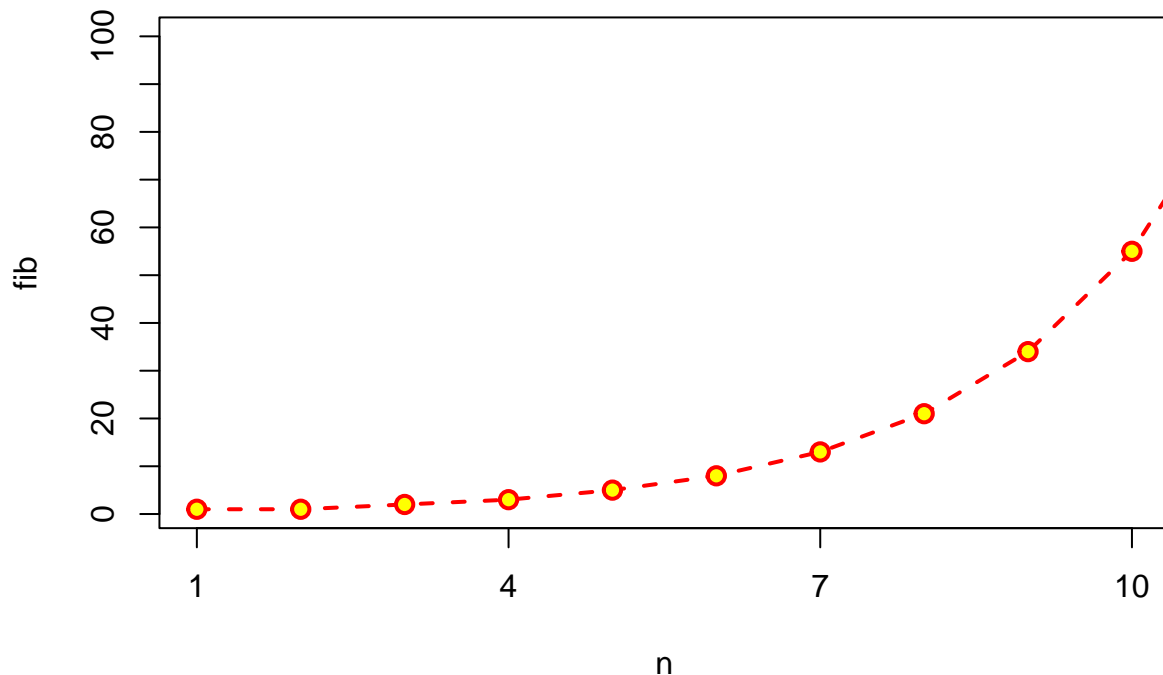
```
par(mfrow = c(1,1))
x = (2*(1:20))
y = (-1)^(1:20)*5*(1:20)
plot(x,y, main = "Ejemplo de grafico", pch=8, cex=1, type="b", lty=4, lwd=4, xaxp=c(0,40,2), yaxp=c(-1
```

Ejemplo de grafico



```
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(1,100),  
     xaxp = c(1,10,3), yaxp = c(0,100,10))
```

Fibonacci

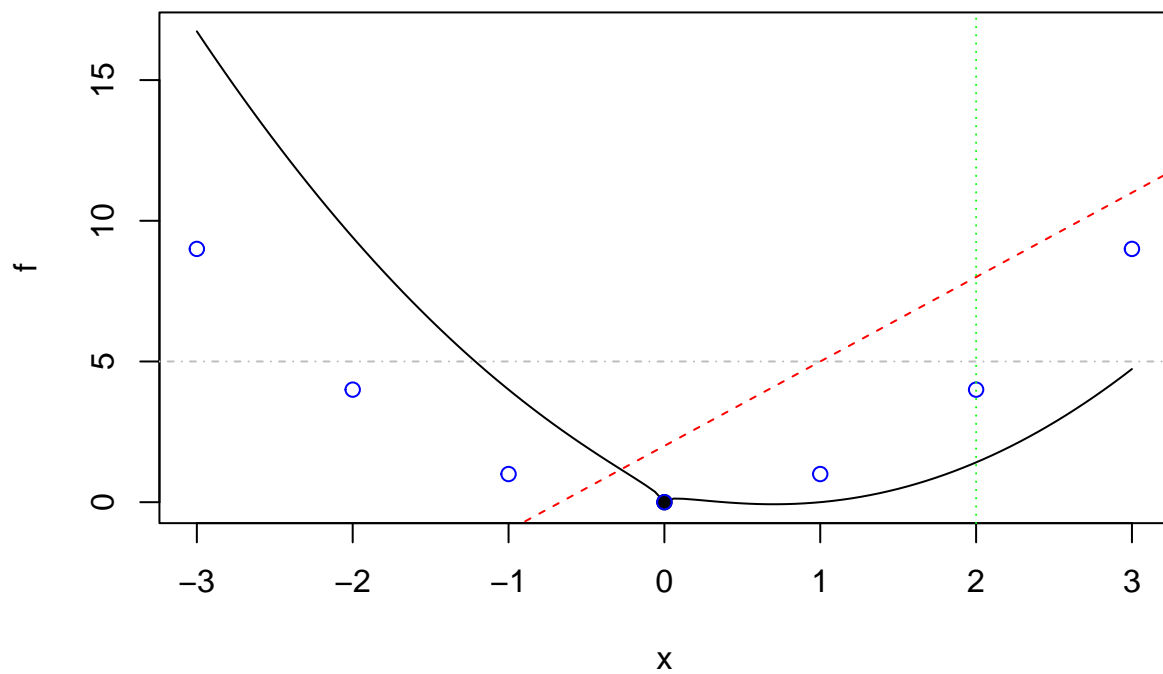


Añadir elementos al grafico.

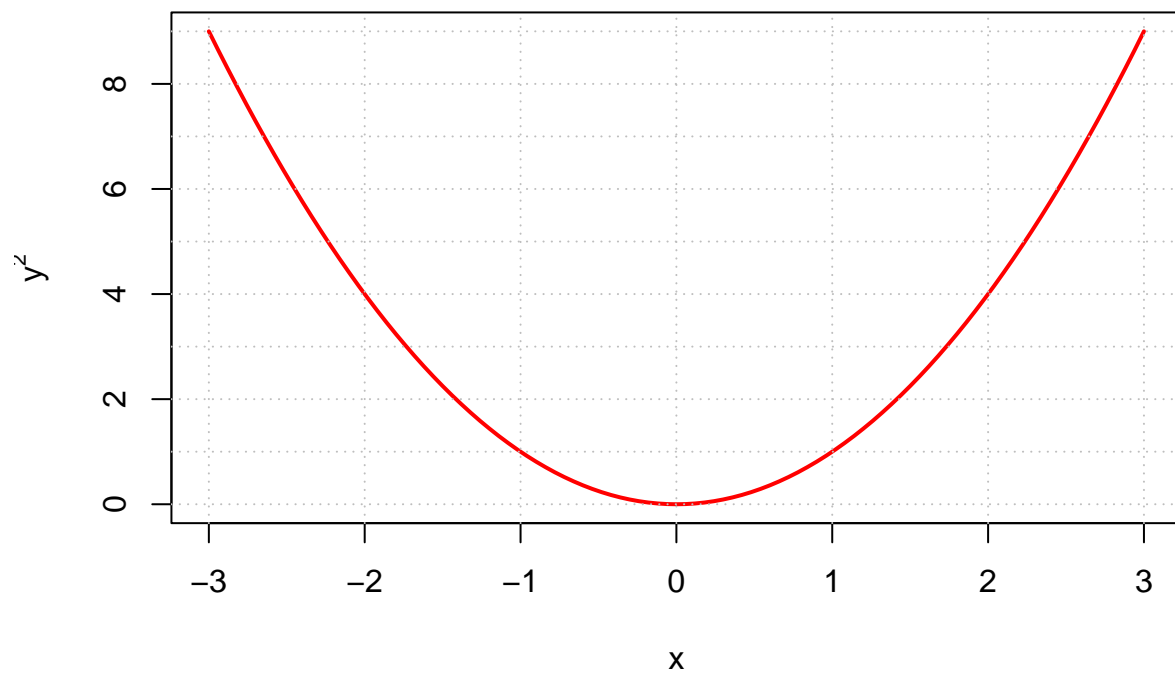
Para hacer cambios en el grafico hay que hacerlo sobre el plot original.

points(x,y) -> añade un punto de coordenadas a un grafico ya existente. abline -> añade una recta a un grafico ya existente. - abline(a,b) -> añade la recta $y = bx + a$ - 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. text(x,y,labels = "...") -> añade en el punto de coordenadas (x,y) el texto especificado. - pos -> permite indicar la posición del texto alrededor de las coordenadas. - 1: abajo - 2: izq - 3: arriba - 4: derecha - 5: sin especificar: centro lines(x, y) -> añade a un grafico existente una linea poligonal que une los puntos(xi ,yi) sucesivos. x,y son vectores numericos curve(curva) -> añadir una curva a la grafica - add=TRUE -> si no, la curva no se añade - Se puede especificar mediante una expresion algebraicaa con variable x, o mediante su nombre si la hemos definido antes legend(posicion, legend = ...) -> Añadir una leyenda. legend contiene el vector de nombres entre comillas. - "bottom" / "bottomright" / "bottomleft" - "top" / "topright" / "topleft" - "center" / "right" / "left"

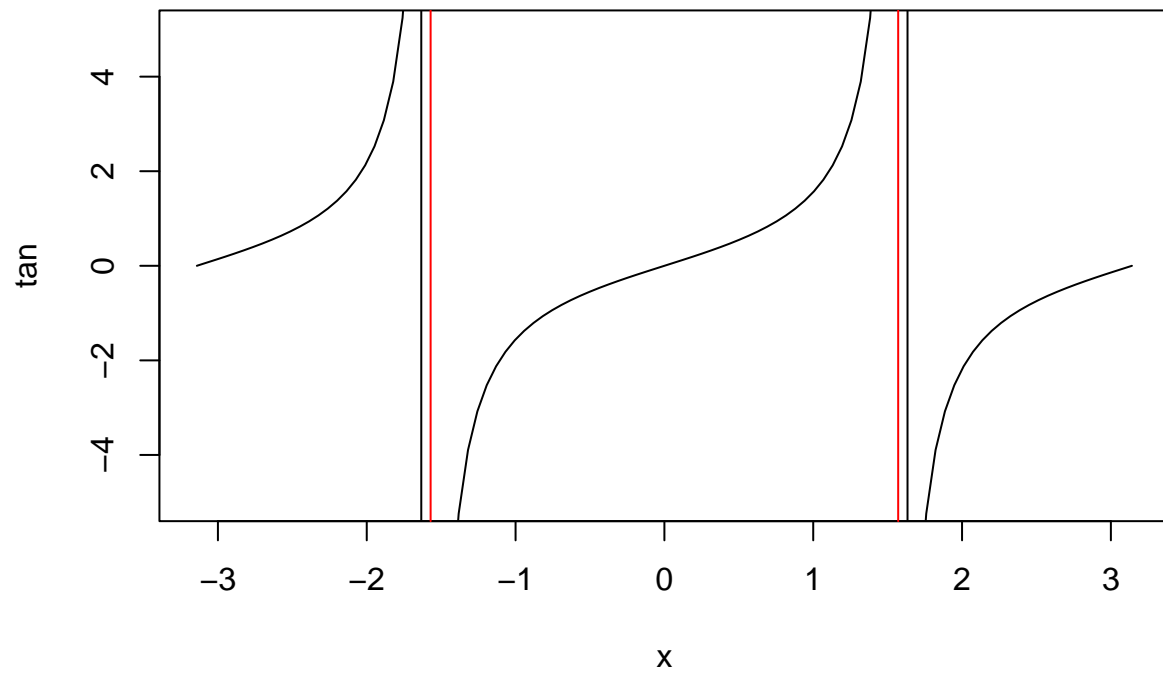
```
f <- function(x){
  x^2 -2*x + sqrt(abs(x))
}
plot(f, xlim = c(-3,3))
points(0,0, pch = 19)
points(-3:3, (-3:3)^2, col = "blue")
abline(2,3, lty = "dashed", col = "red")
abline(v = 2, lty = "dotted", col = "green")
abline(h = 5, lty = "dotdash", col = "gray")
```

```
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")
```

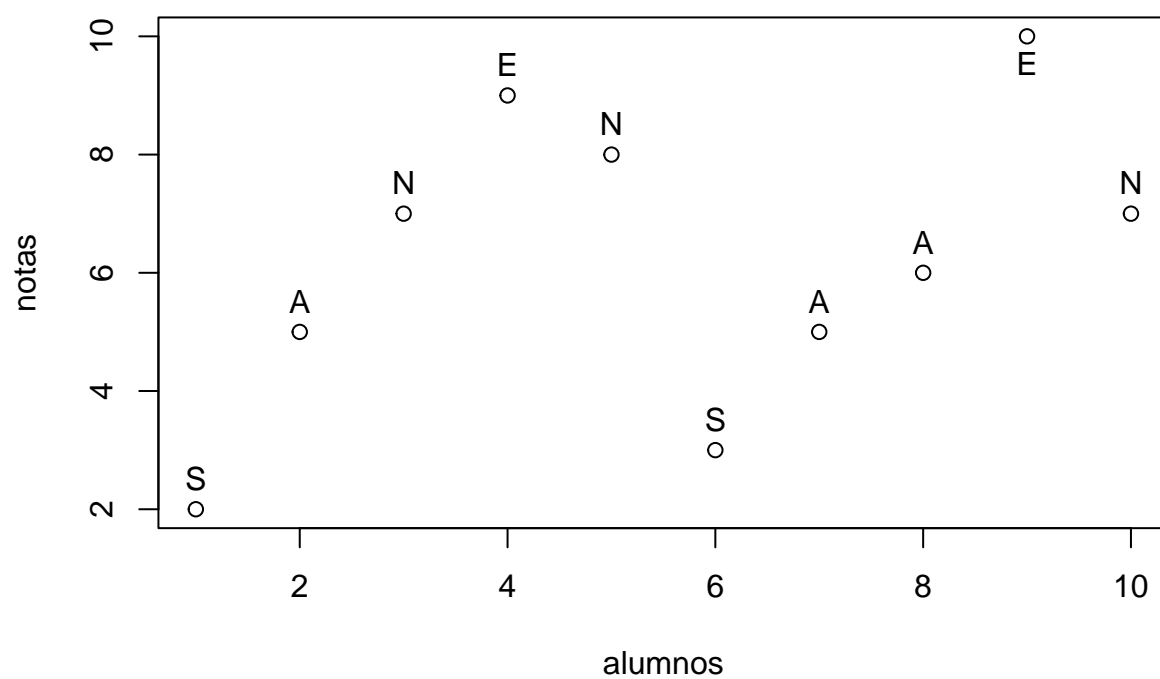


```
plot(tan, xlim=c(-pi,pi), ylim = c(-5,5))  
abline(v = c(-pi/2, pi/2), col = "red")
```

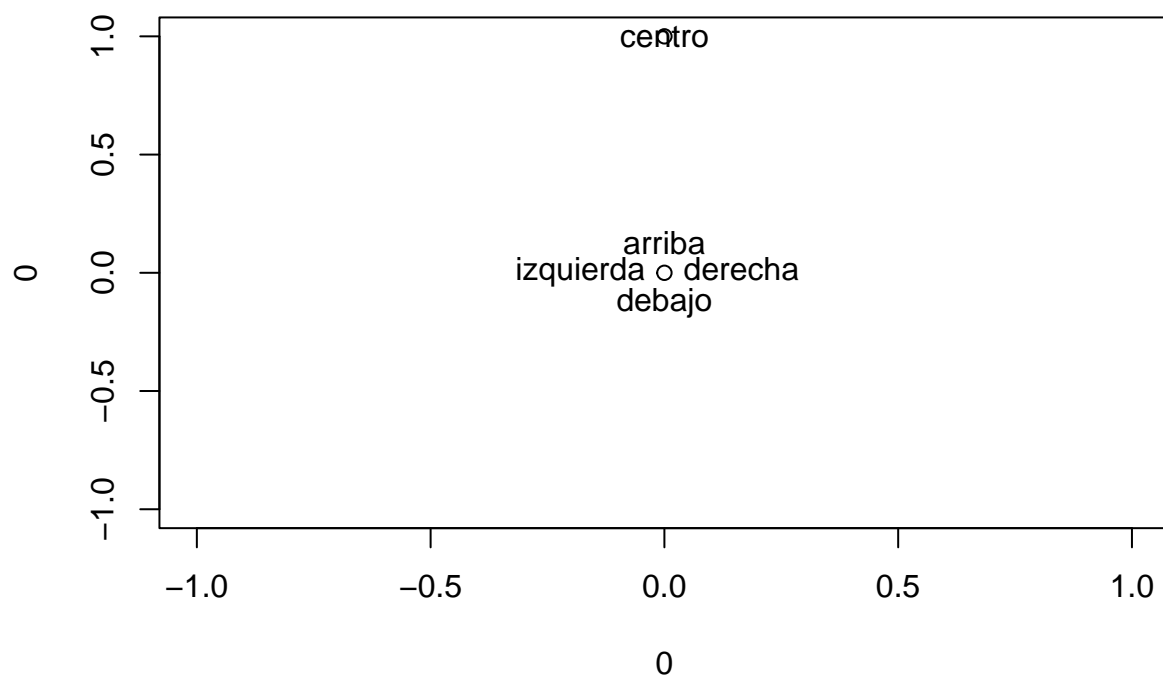


```
# Etiquetas
alumnos = c(1:10)
notas = c(2,5,7,9,8,3,5,6,10,7)
plot(alumnos, notas, main = "Grafico con texto")
text(alumnos, notas, labels = c("S", "A", "N", "E", "N", "S", "A", "A", "E", "N"),
      pos = c(rep(3, times = 8), 1,3))
```

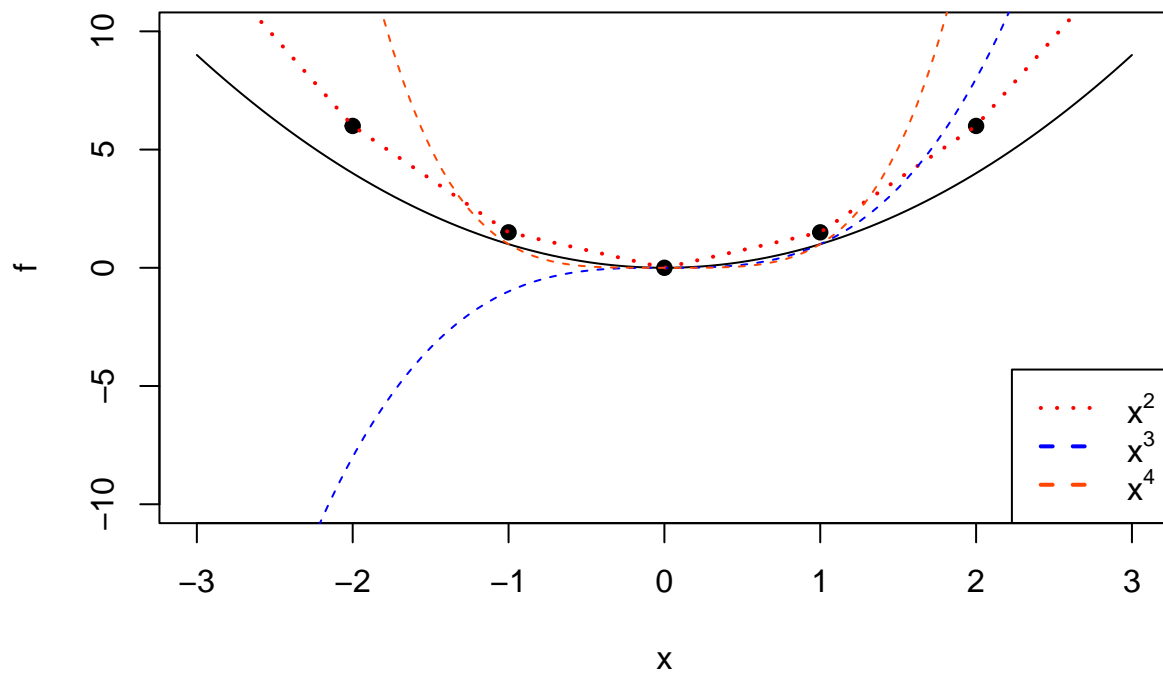
Grafico con texto



```
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")
```



```
f <- function(x){x^2}
plot(f, xlim = c(-3,3), ylim = c(-10,10))
points(-3:3, f(-3:3)*1.5, pch=19)
lines(-3:3, f(-3:3)*1.5, lwd = 2, lty="dotted", col = "red")
curve(x^3, lty="dashed", col = "blue", add = TRUE)
curve(x^4, lty="dashed", col = "orangered", add = TRUE)
legend("bottomright",
      legend = c(expression(x^2), expression(x^3), expression(x^4)),
      lwd = 2,
      col = c("red", "blue", "orangered"),
      lty = c("dotted", "dashed", "dashed")
    )
```



Formula matemática para dibujar un corazón

```
dat<- data.frame(t=seq(0, 2*pi, by=0.1) )
xhrt <- function(t) 16*sin(t)^3
yhrt <- function(t) 13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t)
dat$y=yhrt(dat$t)
dat$x=xhrt(dat$t)
with(dat, plot(x,y, type="l"))
```

