Gráficos

Oscar Gerardo Hernández Martínez

21/7/2019

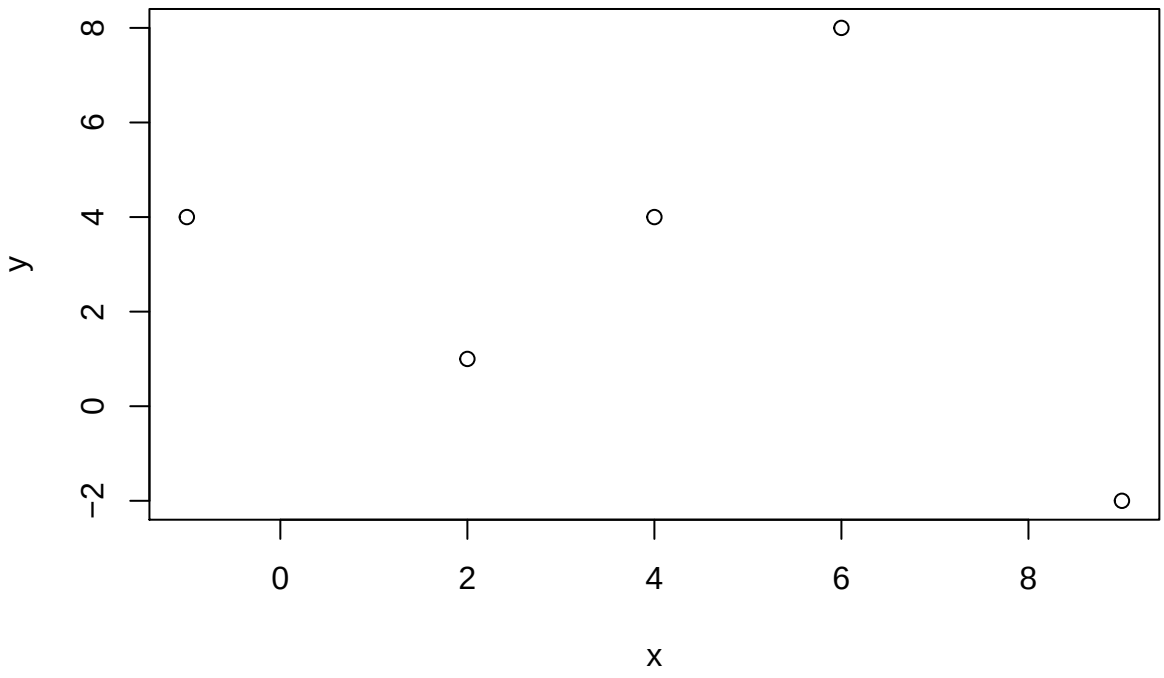
Gráficos con la función `plot`

Gráfico básico de puntos

- **plot(x,y):** Para dibujar un gráfico básico de puntos siendo x, y vectores numéricos.
  - **plot(x) = plot(1:length(x),x)** -> En este caso, x representa el valor de la coordenada y.
- **plot(x,función):** Para dibujar el gráfico de una función.

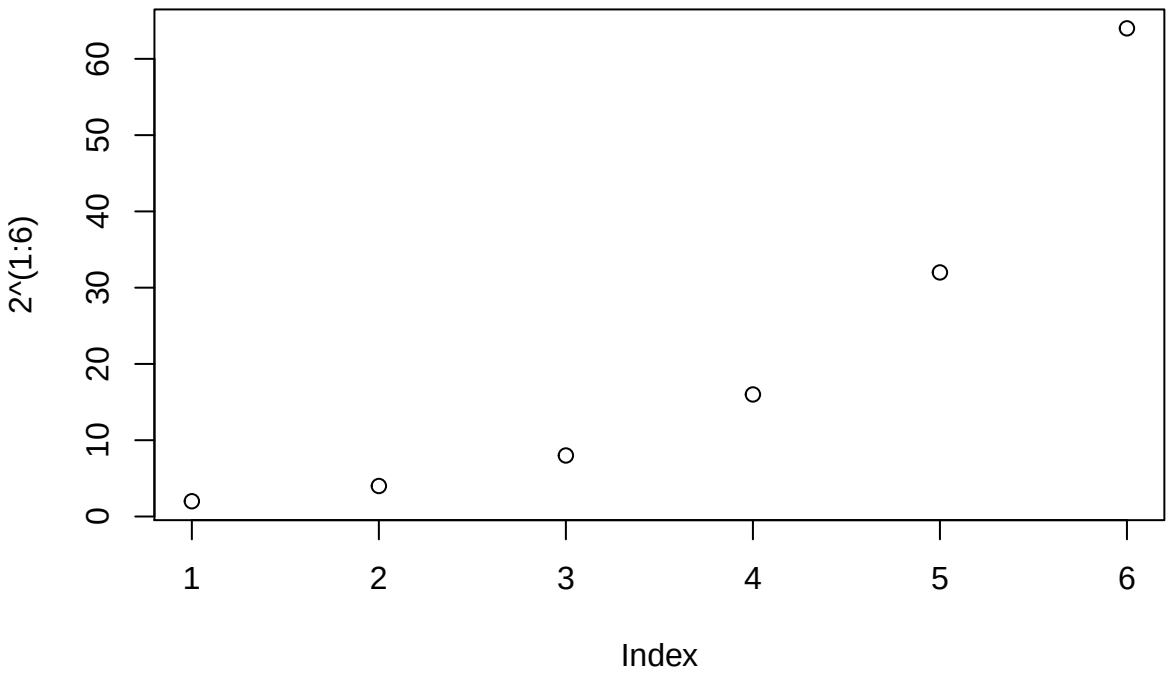
Ejemplo:

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



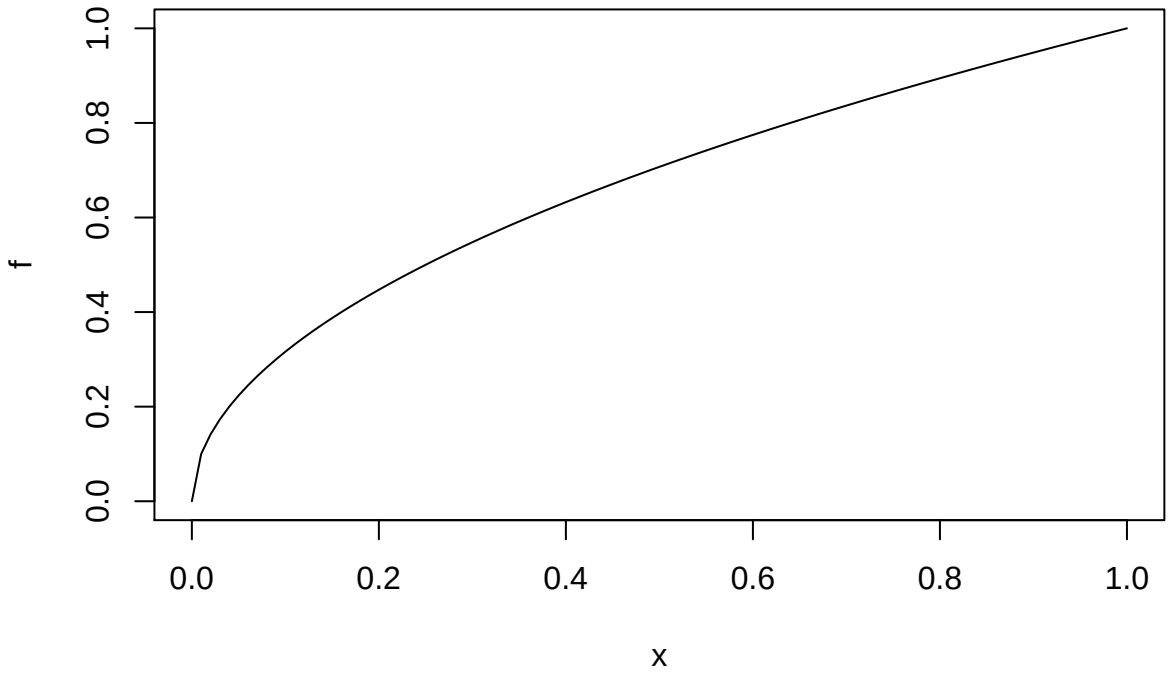
Si no incorporamos vector `y`, `R` nos va a tomar el parámetro `x` como si fuese el vector de datos `y`.

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



Si queremos representar una función:

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



En caso de querer añadir un pie de página a la gráfica se debe agregar dentro de la chunk el parámetro **fig.caption** así: {r, fig.caption="Texto"} y tambien agregar el código **fig.align='center'** quedando como resultado: {r, fig.caption="Texto", fig.align='center'}

Parámetros de la función plot()

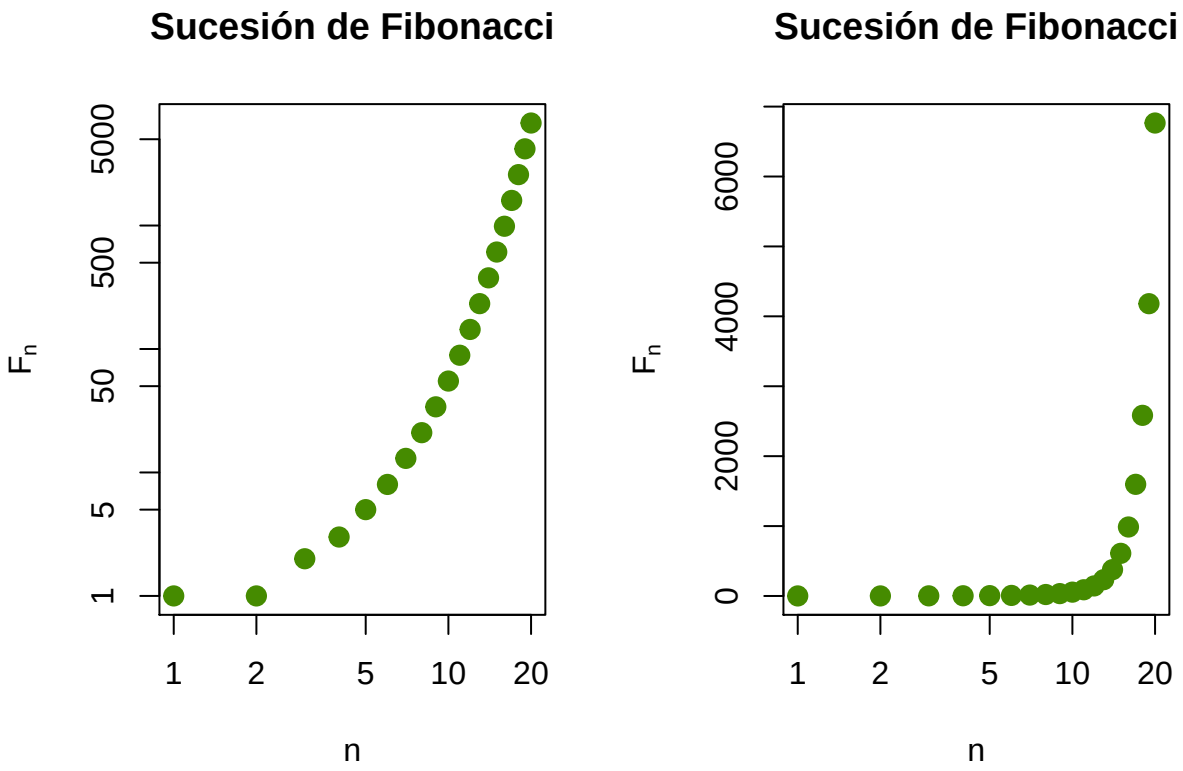
- **log:** Para indicar que queremos el gráfico en escala logarítmica.
- **main("título"):** Para poner título al gráfico. Si en vez de un texto se desea poner una expresión matemática, se tiene que utilizar la función **expression()**.
- **xlab("etiqueta"):** Para poner etiqueta al eje X.
- **ylab("etiqueta"):** Para poner etiqueta al eje Y.
- **pch=n:** Para elegir el símbolo de los puntos. n=0,1,...,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. Gama de colores <–¡PRESIONA AQUÍ PARA ACCEDER A LOS CÓDIGOS DE LOS COLORES!

Parámetros

```
n = 1:20
fib = (1/sqrt(5))*((1+sqrt(5))/2)^n - (1/sqrt(5))*((1-sqrt(5))/2)^n #Fórmula
# para obtener la sucesión de Fibonacci
fib
```

```
##  [1]    1    1    2    3    5    8   13   21   34   55   89  144  233  377
## [15]  610  987 1597 2584 4181 6765
```

```r
par(mfrow = c(1,2)) #Con esto puedo hacer que se ordenen
#mis gráficos en una fila y dos columnas.
plot(fib, xlab = "n", ylab = expression(F[n]),
     main = "Sucesión de Fibonacci", pch = 20, cex = 2,
     col = "chartreuse4", log = "xy")
plot(fib, xlab = "n", ylab = expression(F[n]),
     main = "Sucesión de Fibonacci", pch = 20, cex = 2,
     col = "chartreuse4", log = "x")
```



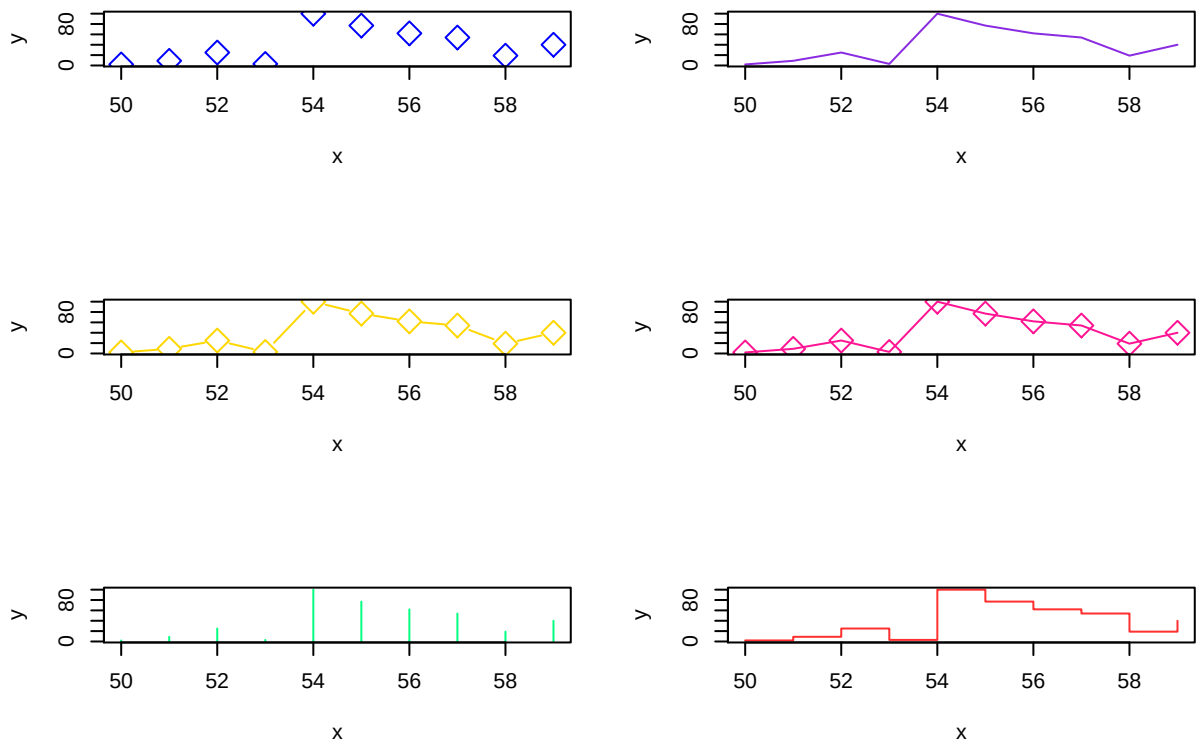

```r
par(mfrow = c(1,1)) #Lo regreso para que los
#gráficos se muestren en pantalla.
```

Más parámetros de la función plot()

- **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.
- **lty:** para especificar el tipo de línea:
  - "solid" : 1: línea continua (valor por defecto).
  - "dashed" : 2: línea discontinua.

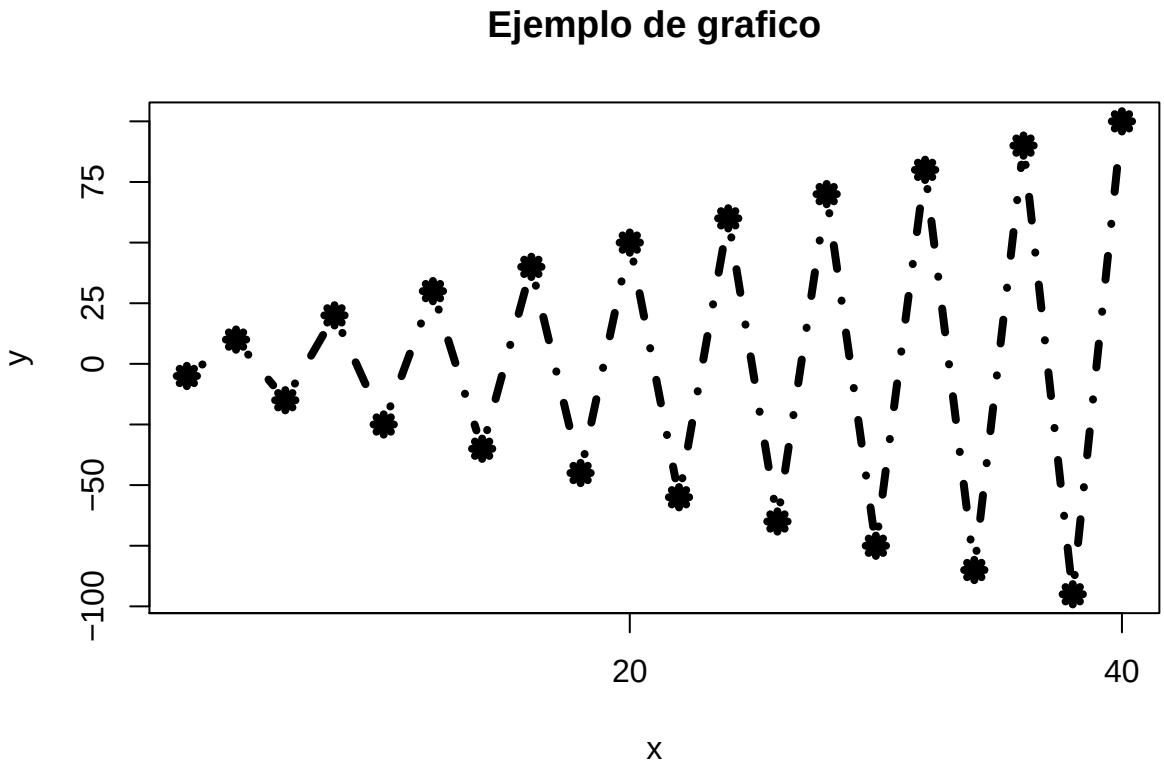
  - "dotted" : 3: línea de puntos.
  - "dotdashed" : 4: línea que alterna puntos y rayas.
- **lwd:** para especificar el grosor de las líneas.
- **xlim:** para modificar el rango del eje X.
- **ylim:** para modificar el rango del eje Y.
- **xaxp:** para modificar posiciones de las marcas en el eje X.
- **yaxp:** para modificar posiciones de las marcas en el eje Y.

```
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")
```



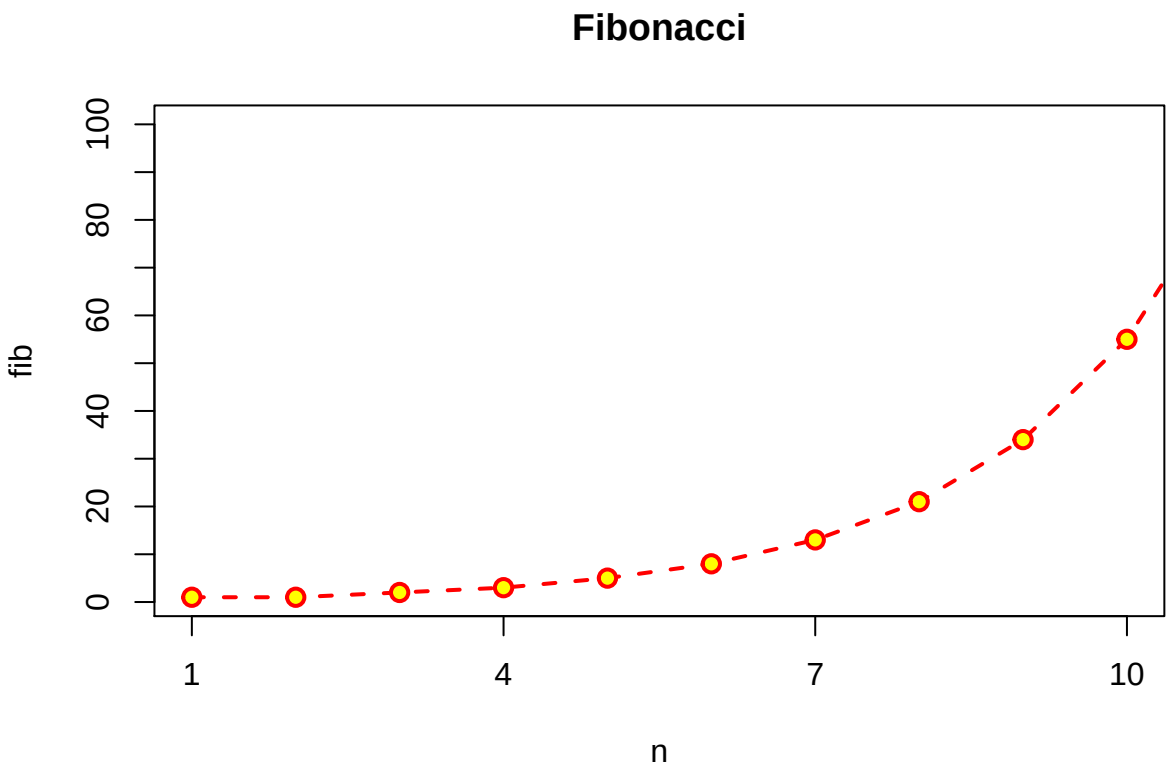

```
par(mfrow = c(1,1))
```

```
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(-100,100,8)) #Entre los números 0 y 40 quiero
```




```
#dos divisiones y entre los números -100 y 100
#quiero 8 divisiones.
```

```
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))
```

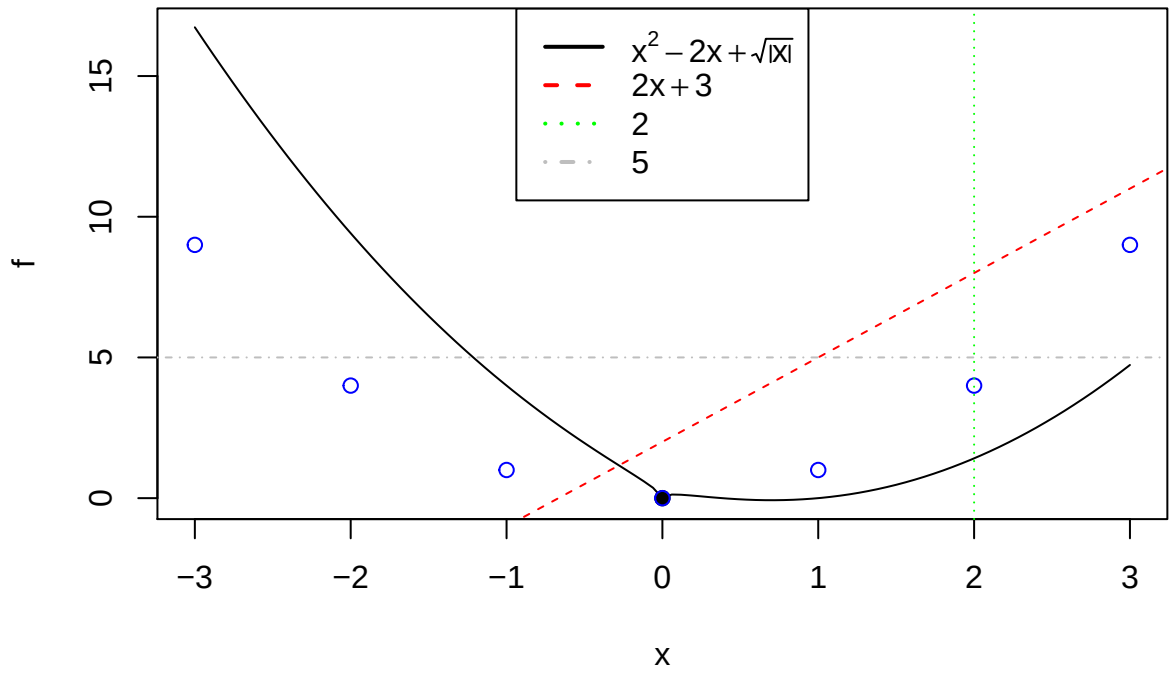


Añadir elementos al gráfico

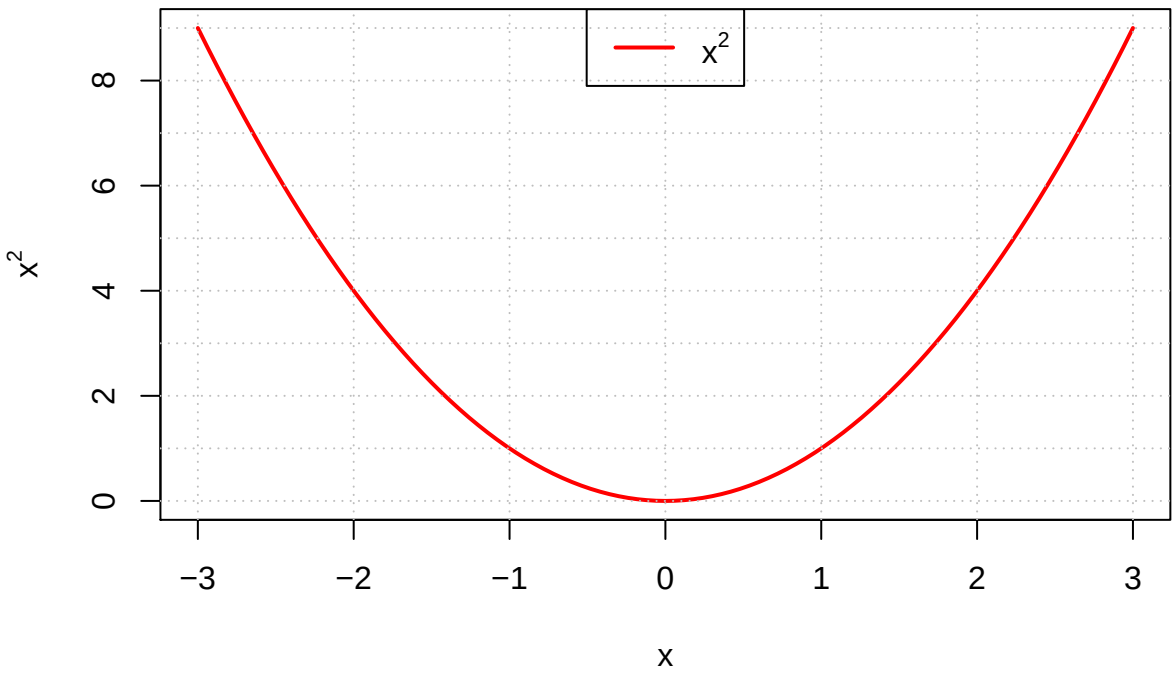
- **points(x,y):** Añade un punto de coordenadas `(x,y)` a un gráfico ya existente.
- **abline:** Para añadir una recta a un gráfico ya existente.
  - **abline(a,b):** Añade la recta $y = ax + b$.
  - **abline(v = x0):** Añade la recta vertical $x = x_0$. `v` puede estar asignado a un vector.
  - **abline(h = y0):** Añade la recta horizontal $y = y_0$. h puede estar asignado a un vector.
- **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)
- **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.
  - *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.
- **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.
- **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.

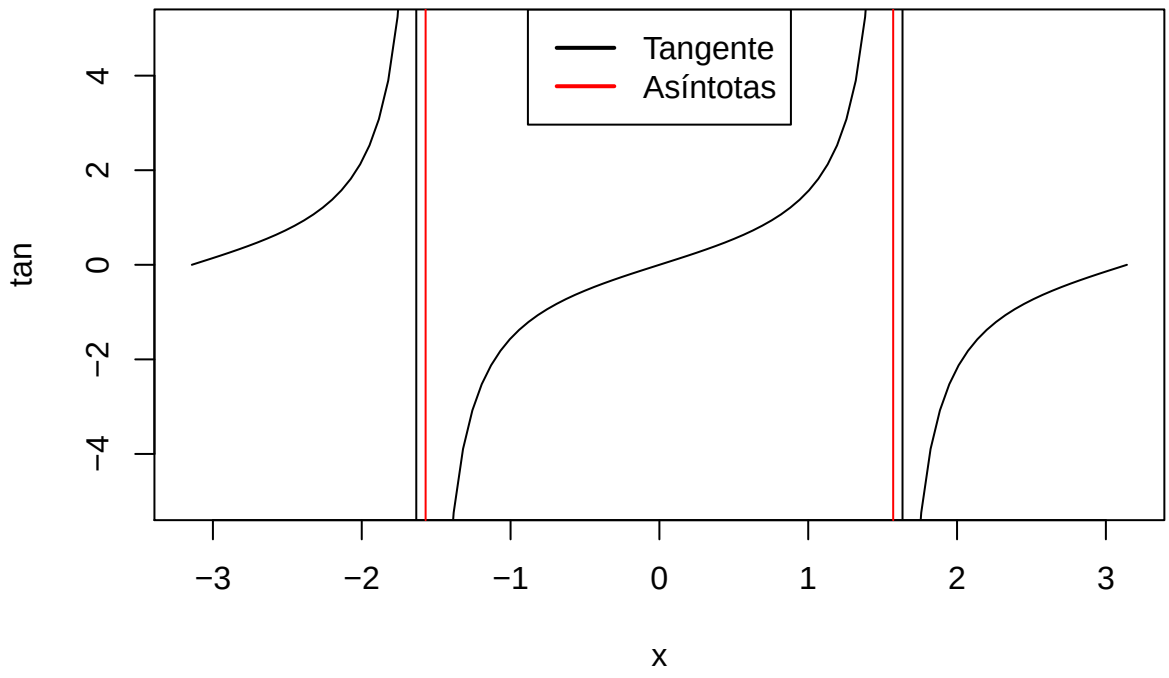
```
f <- function(x){
  x^2 - 2*x + sqrt(abs(x))
}
plot(f, xlim = c(-3,3))
points(0,0, pch = 19)
points(-3:3, (-3:3)^2, col = "blue")
abline(2,3, lty = "dashed", col = "red")
abline(v = 2, lty = "dotted", col = "green")
abline(h = 5, lty = "dotdash", col = "gray")
legend("top",
       legend = c(expression(x^2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



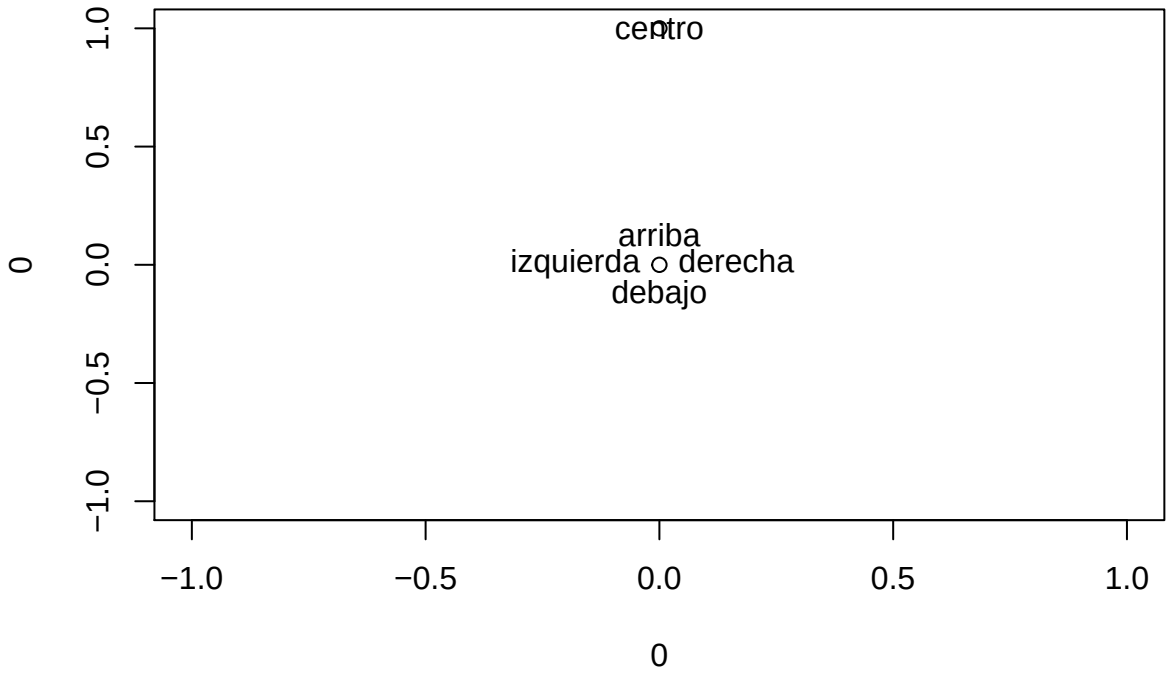
```r
f <- function(x){
  x^2
}
plot(f, xlim = c(-3,3), col = "red", lwd = 2,
     ylab = expression(x^2), xlab = "x")
abline(h = 0:9, v = -3:3, lty = "dotted", col = "gray")
legend("top", legend = expression(x^2),
       col = "red", lwd = 2, lty = 1)
```



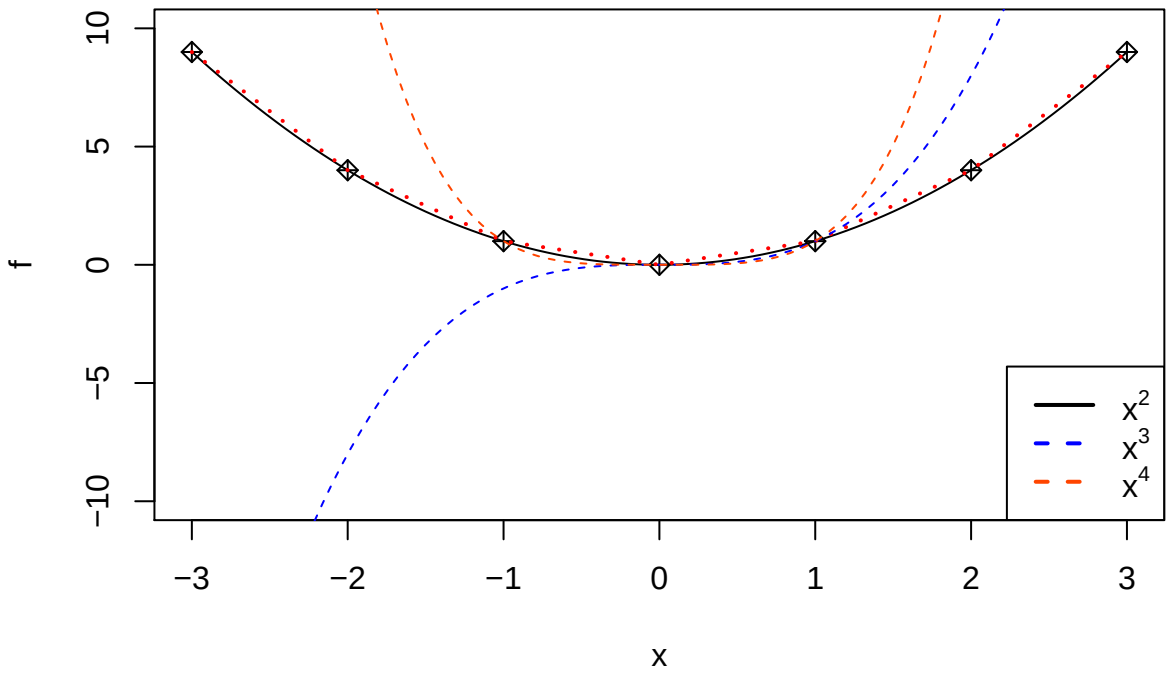

```r
plot(tan, xlim = c(-pi, pi), ylim = c(-5,5))
abline(v = c(-pi/2, pi/2), col = "red")
legend("top",
       legend = c("Tangente", "Asíntotas"),
       col = c("black", "red"), lwd = 2, lty = 1)
```



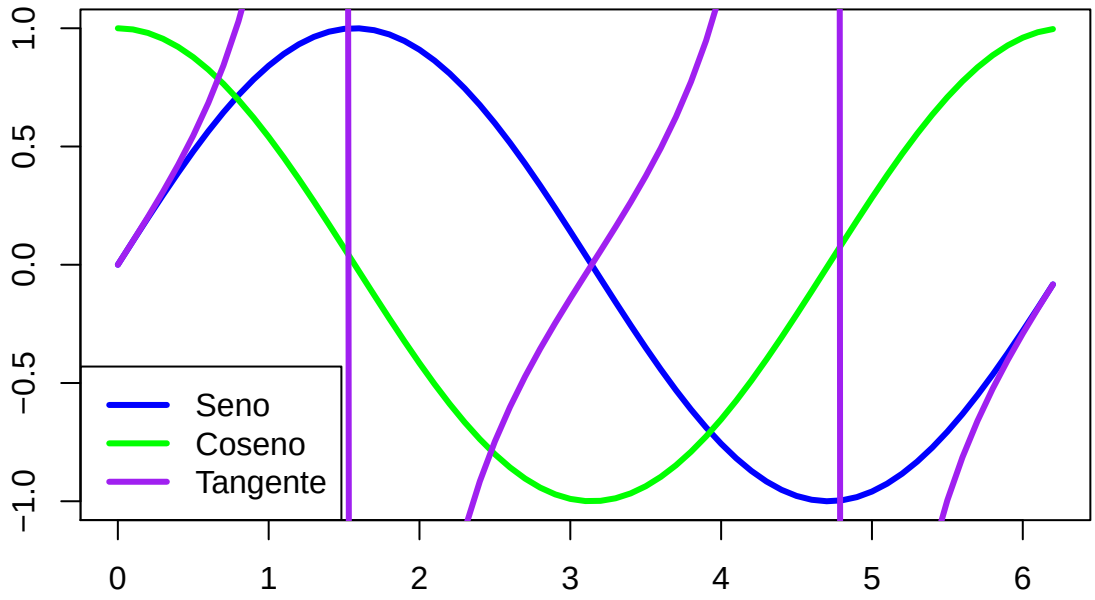

```
plot(0,0)
text(0,0, labels = "debajo", pos = 1)
text(0,0, labels = "izquierda", pos = 2)
text(0,0, labels = "arriba", pos = 3)
text(0,0, labels = "derecha", pos = 4)
points(0,1)
text(0,1, labels = "centro")
```



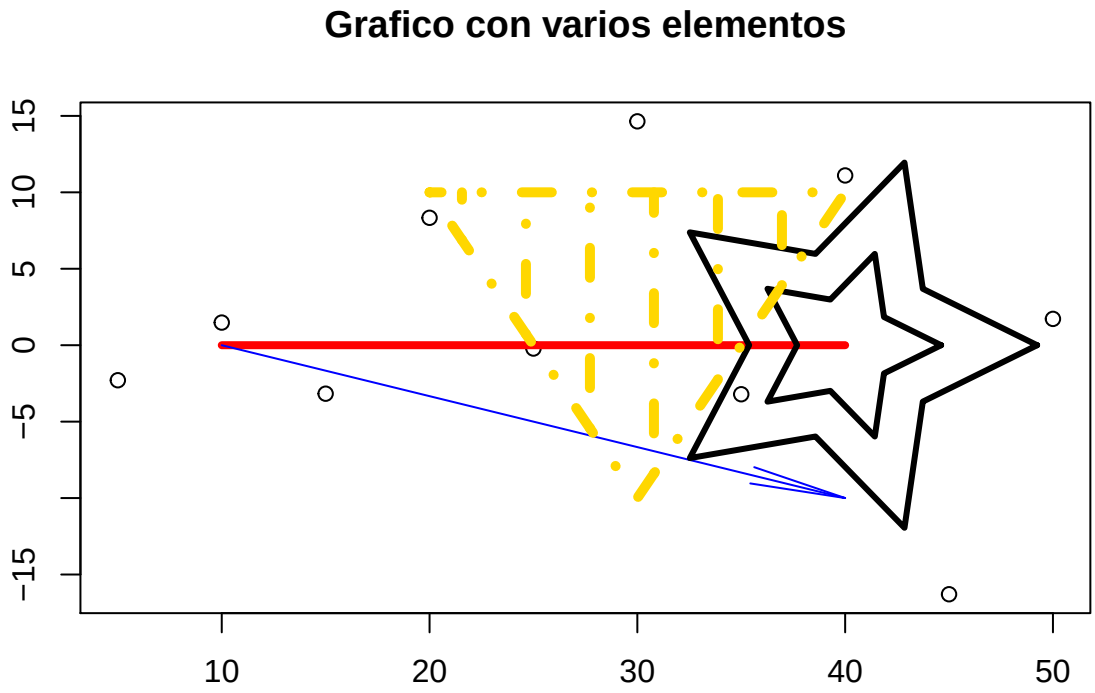

```r
f <- function(x){x^2}
plot(f, xlim = c(-3,3), ylim = c(-10,10))
points(-3:3, f(-3:3), pch = 9)
lines(-3:3, f(-3:3), lwd = 2, lty = "dotted", col = "red")
curve(x^3, lty = "dashed", col = "blue", add = TRUE)
curve(x^4, lty = "dashed", col = "orangered", add = TRUE)
legend("bottomright",
       col = c("black", "blue", "orangered"),
       legend = c(expression(x^2), expression(x^3),
                  expression(x^4)),
       lwd = 2,
       lty = c("solid", rep("dashed", times=2))
       )
```




```
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")
```




```
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) #Genera una línea
#(segmento) de color rojo que empieza en (10,0) y
#termina en (40,0)
arrows(10,0,40,-10, col = " blue", length = 0.5, angle = 5, code = 2)
#Genera una línea color azul con inclinación 5
#Que empieza en (10,0) y termina en (40,-10)
symbols(40,0,stars = cbind(1,.5,1,.5,1,.5,1,.5,1,.5), add = TRUE, lwd = 3, inches = 0.5) #Dibuja una es
#el 100% de su tamaño y el 50% de su tamaño
#En este ejemplo la colocamos al 50% del tamaño original
symbols(40,0,stars = cbind(1,.5,1,.5,1,.5,1,.5,1,.5), add = TRUE, lwd = 3)
#Dibuja una estrella cuyos vértices alternan entre
#el 100% de su tamaño y el 50% de su tamaño
#En este ejemplo la colocamos al 100% de su tamaño original
polygon(c(20,30,40),c(10,-10,10), col = "gold", density = 3, angle = 90, lty = 4,
        lwd = 5)
```




```
#Genera un triángulo color or
```