Gráficos

Curso de Estadística Descriptiva

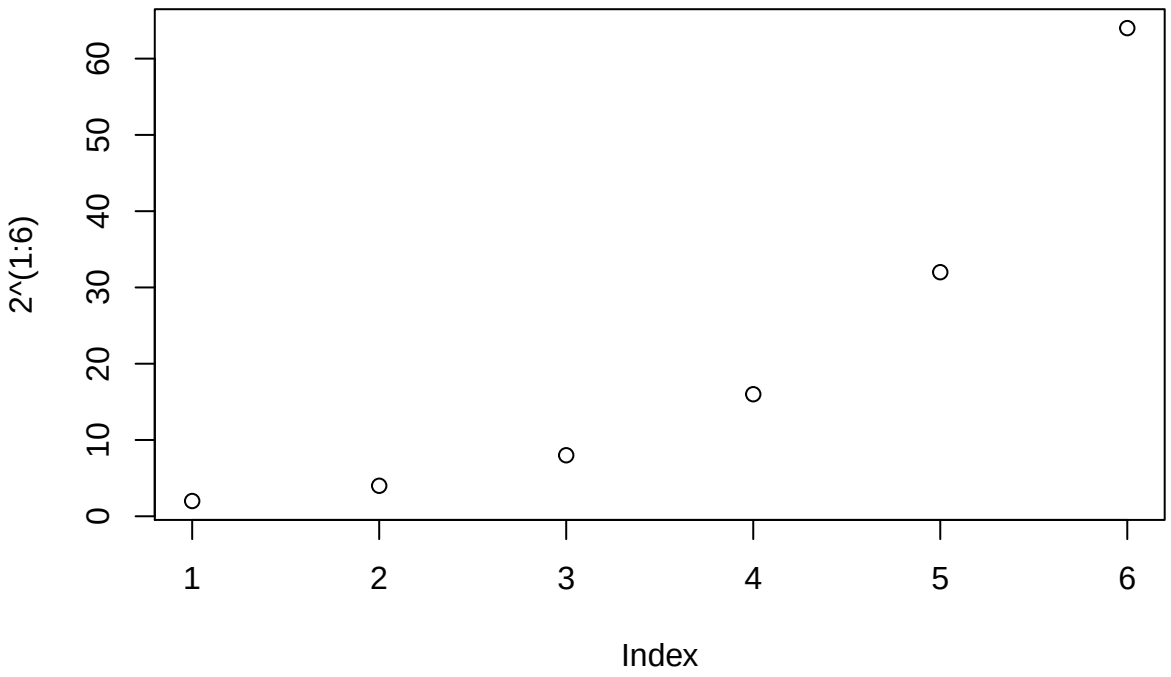
8/4/2020

Gráficos con la función `plot`

```
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)`

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



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

```
f = function(x) { sqrt(x) }

plot(f)
```

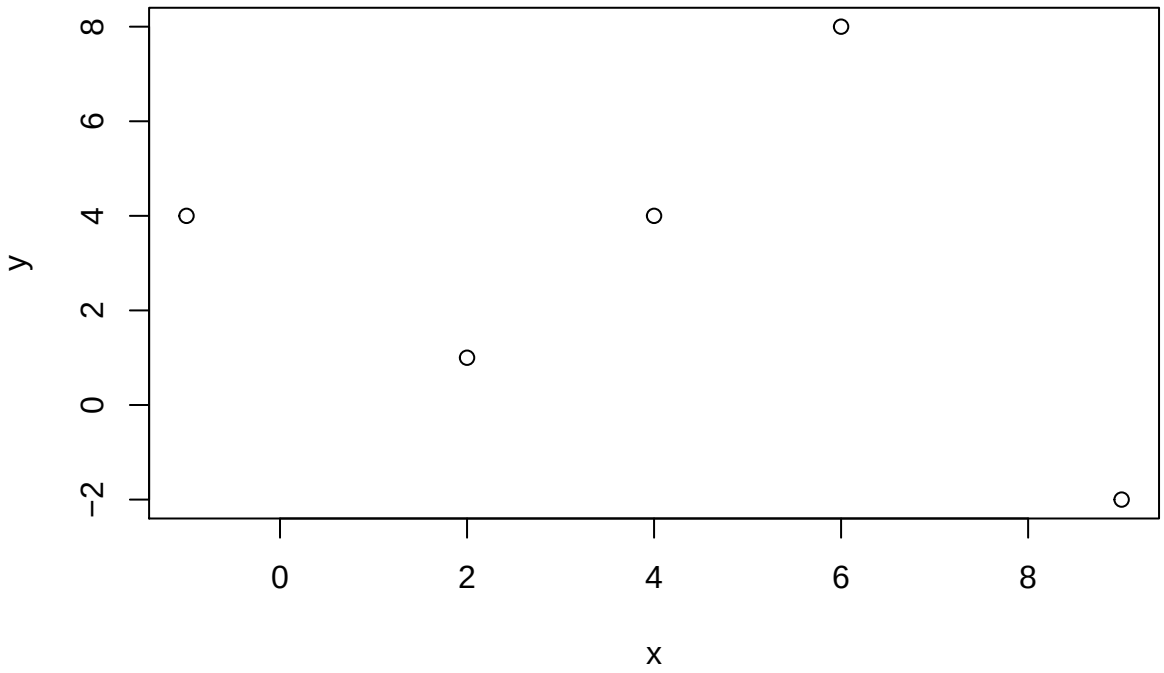
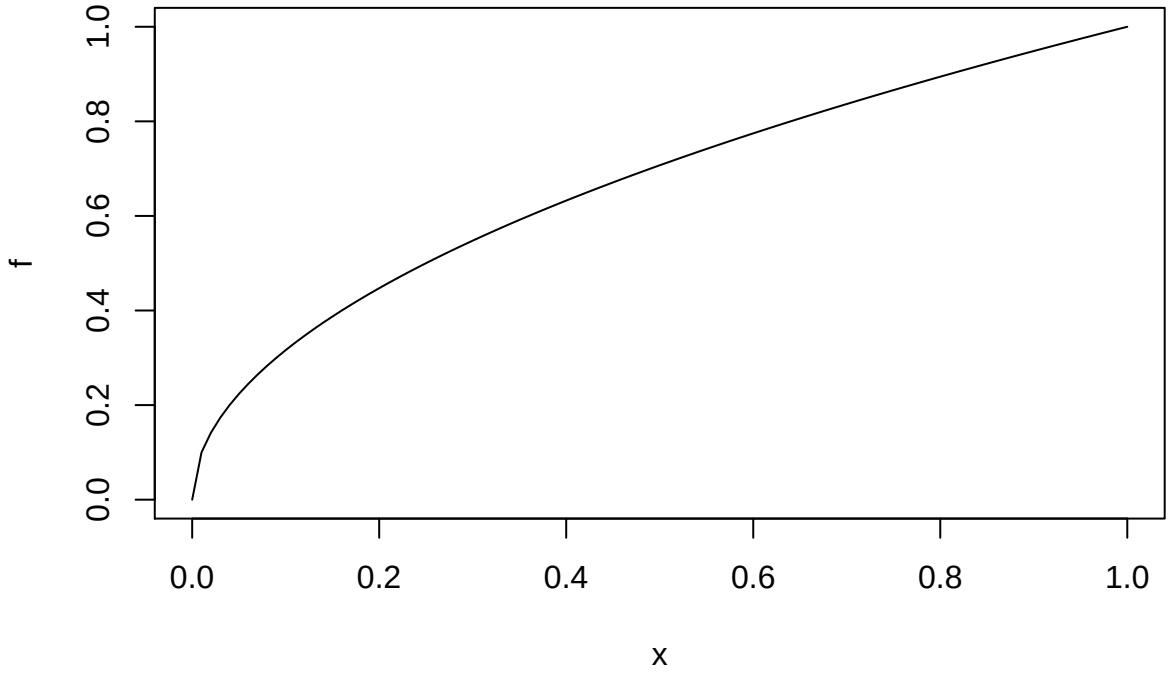
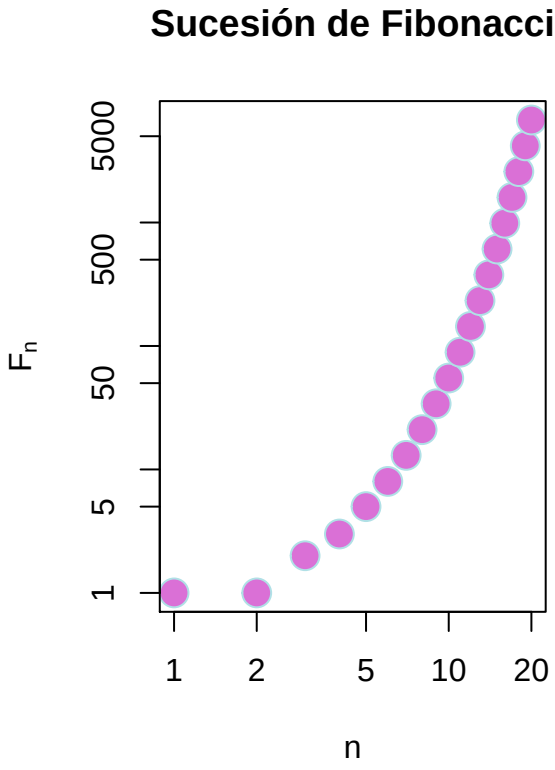


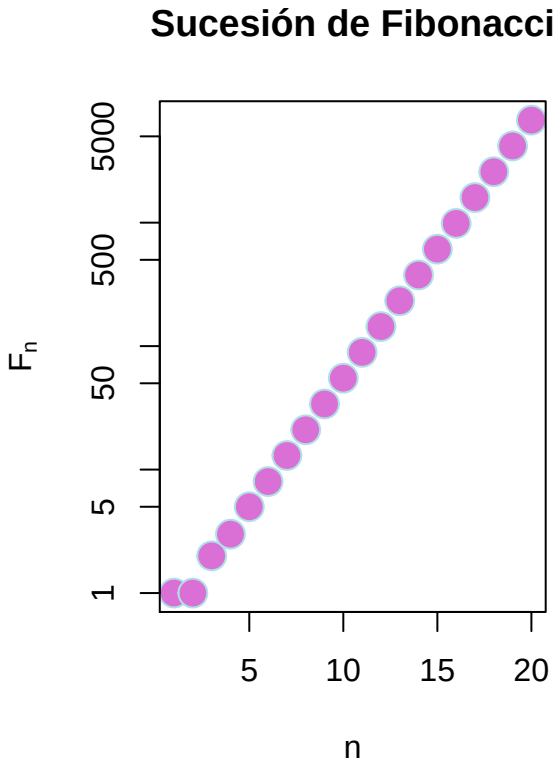
Figure 1: Gráfico básico explicando el uso de plot



Parámetros

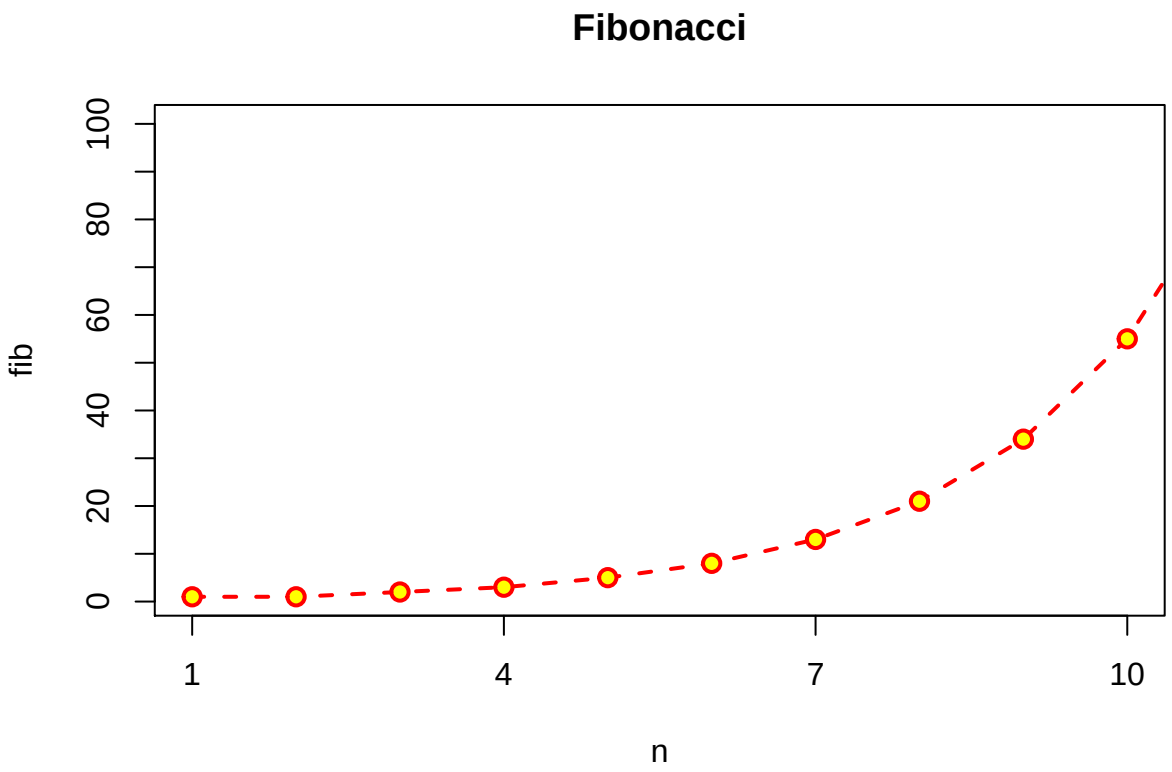
```
##  [1]    1    1    2    3    5    8   13   21   34   55   89  144  233  377
## [15]  610  987 1597 2584 4181 6765
```





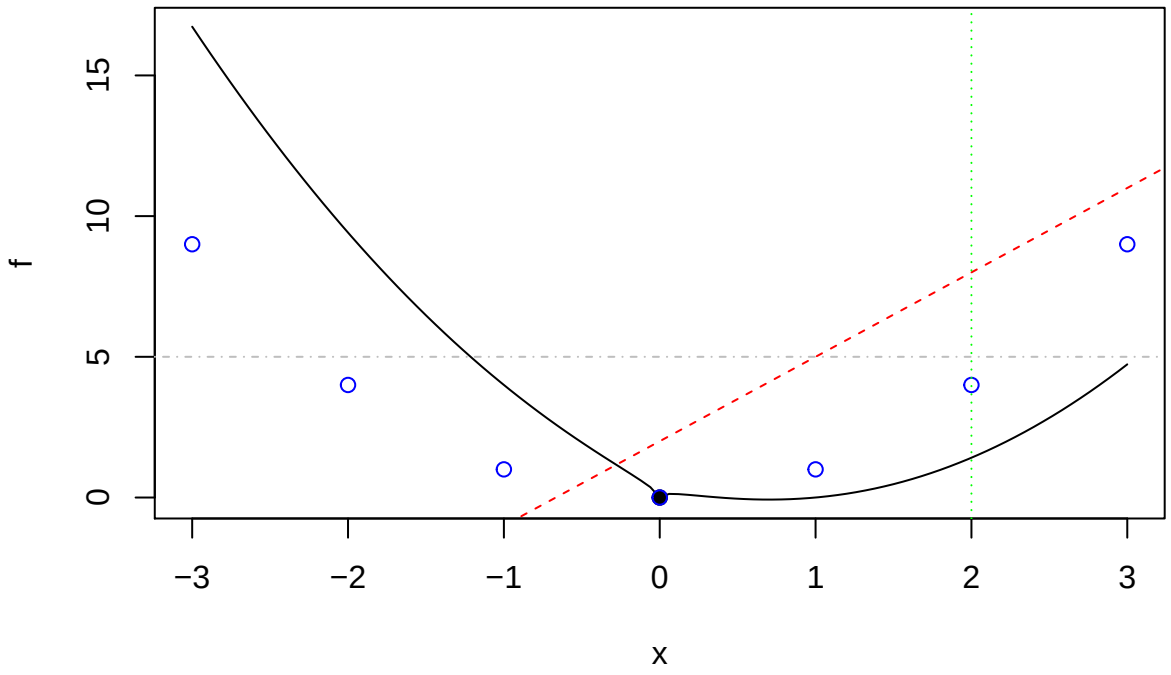

```
plot(n, fib, pch = 21,
     col = "red", bg = "yellow",
     cex = 1.2, main = "Fibonacci",
     type = "o", lty = "dashed",
```

```
     lwd = 2, xlim = c(1, 10),
     ylim = c(1, 100),
     xaxp = c(1, 10, 3), yaxp = c(0, 100, 10))
```

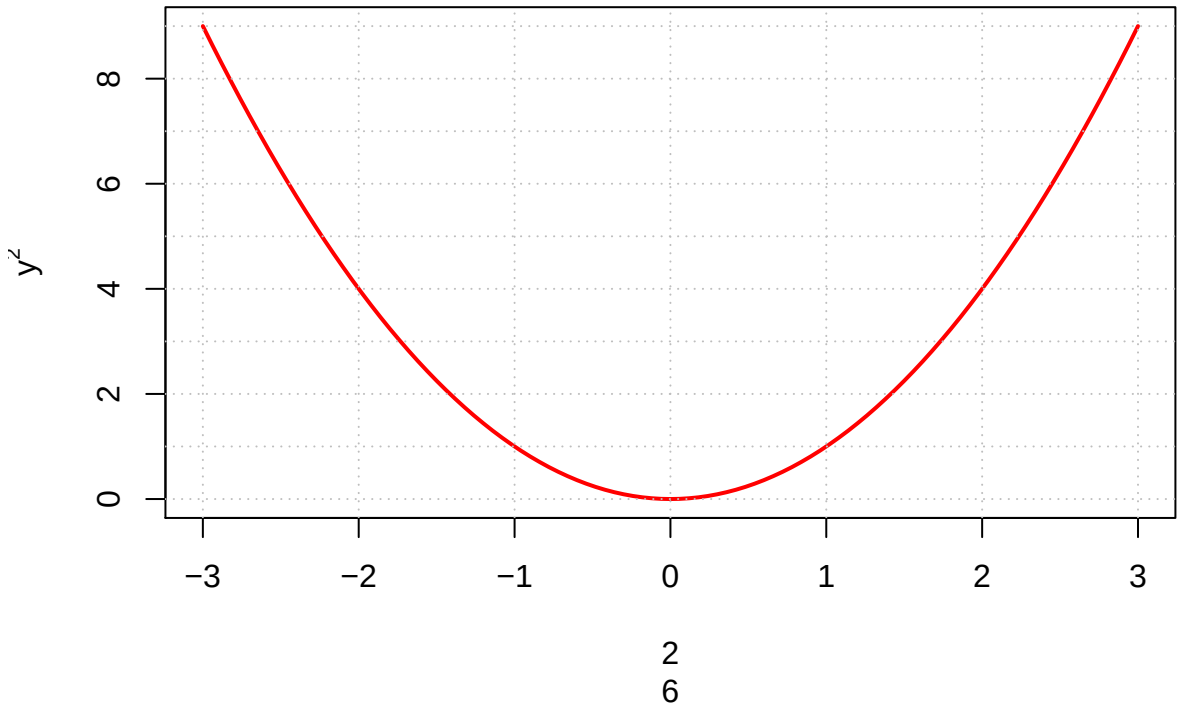


Cómo añadir elementos a un gráfico

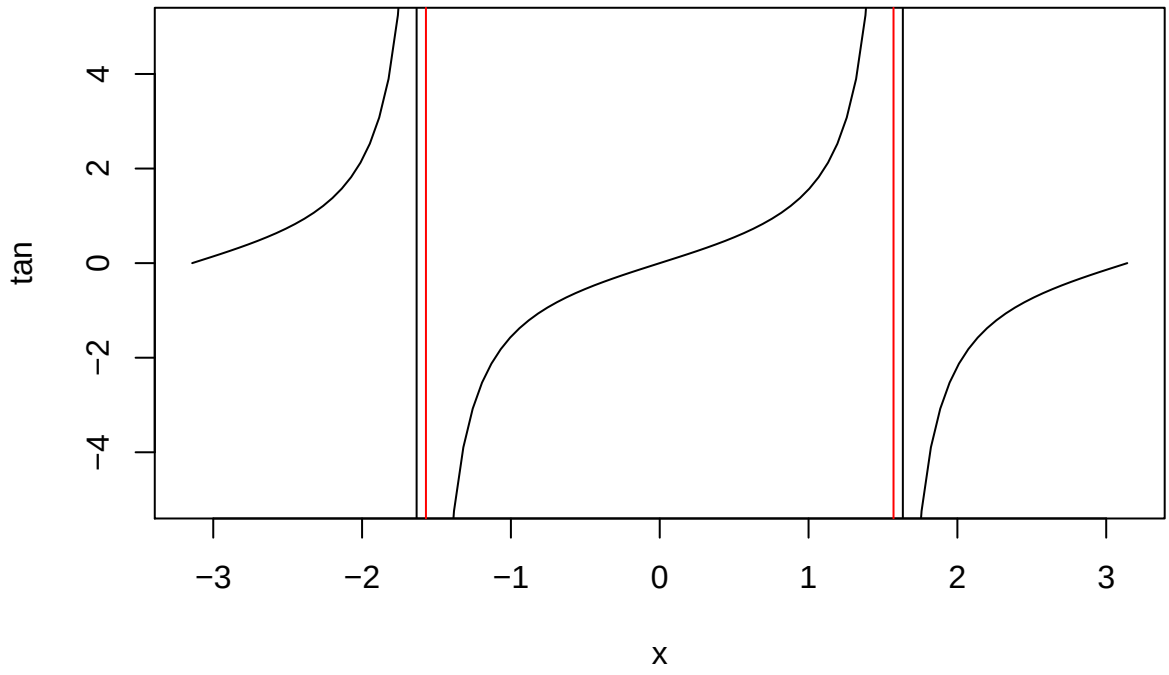
```
f = function(x) {
  x^2 - 2*x + sqrt(abs(x))
}

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 = "gray")
```



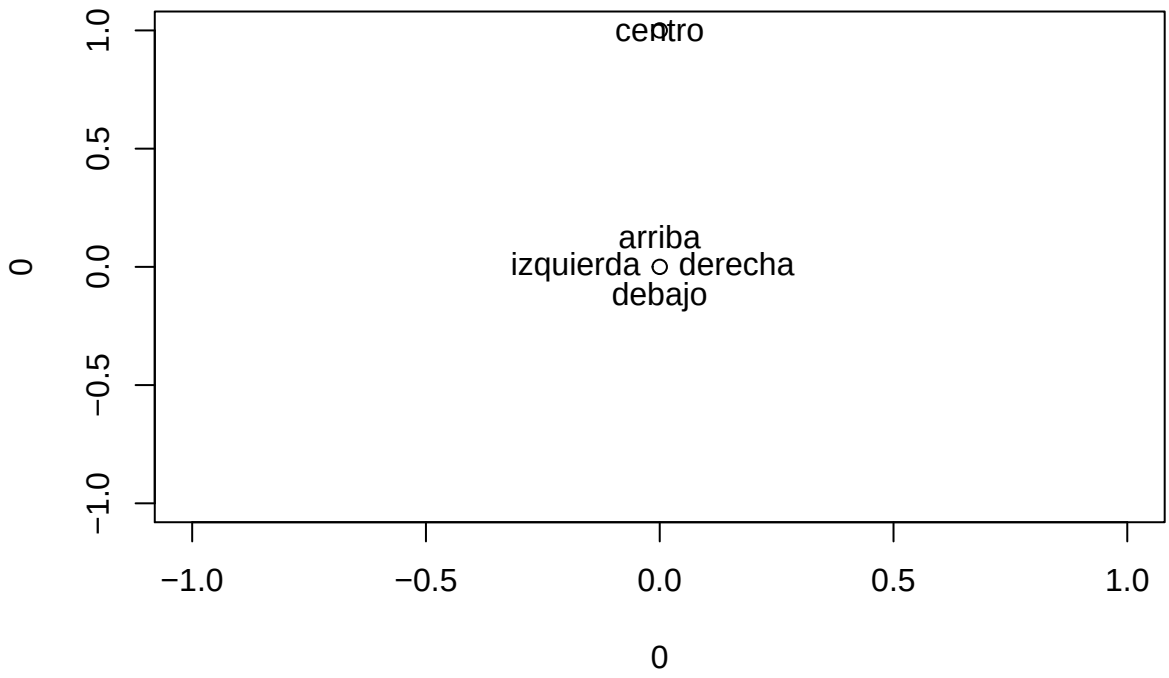

```
f = function(x) {x^2}
  plot(f, xlim = c(-3,3), col = "red", lwd = 2, ylab = expression(y^2), xlab = x)
  abline(h = 0:9, v = -3:3, lty = "dotted", col = "gray")
```



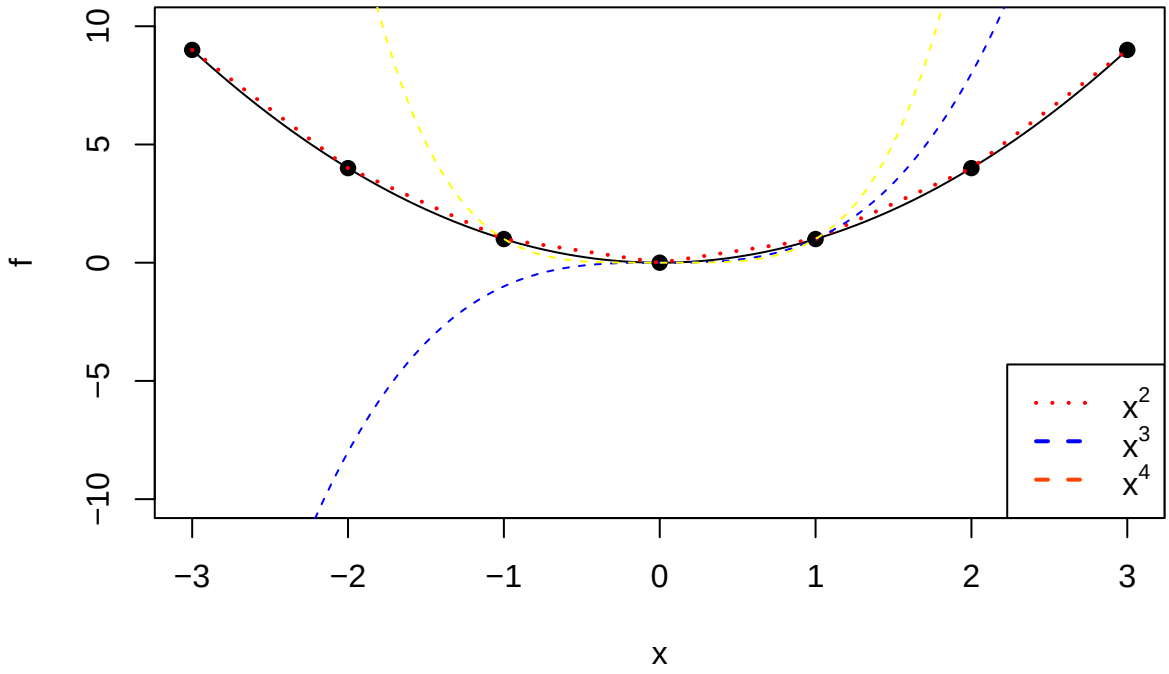

```
plot(tan, xlim = c(-pi, pi), ylim = c(-5, 5))
abline(v = c(-pi/2, pi/2), col = "red")
```




```
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")
```

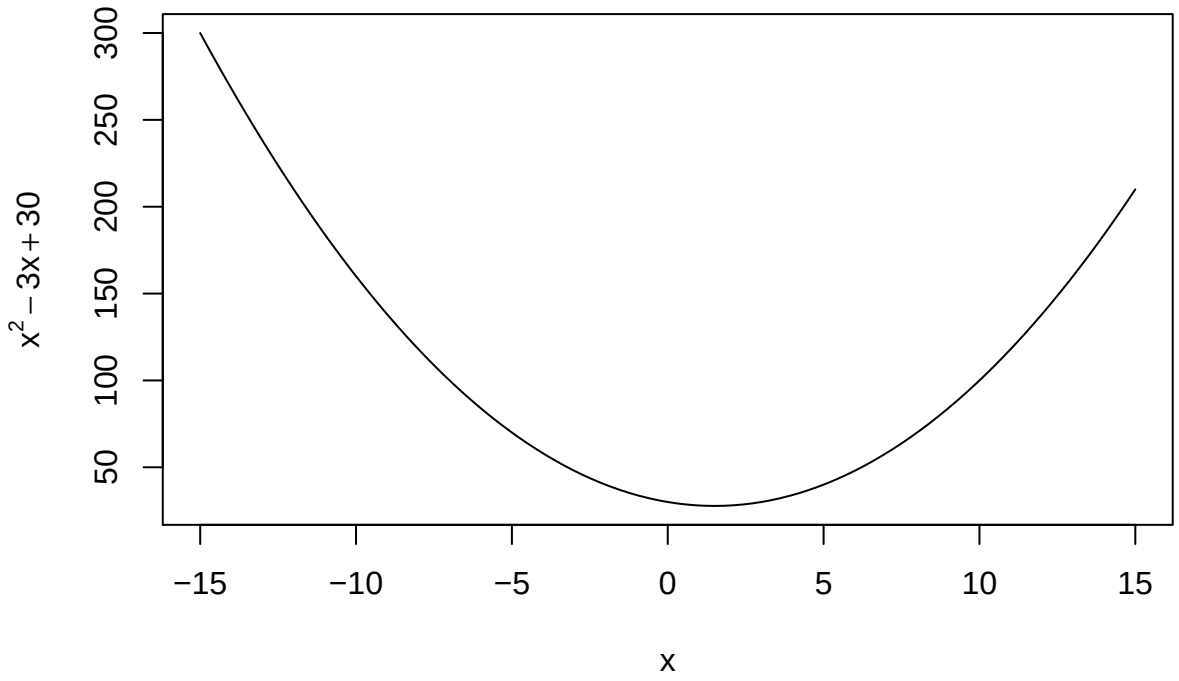


```r
f = function(x){x^2}
plot(f, xlim = c(-3, 3), ylim = c(-10, 10))
points(-3:3, f(-3:3), pch = 19)
lines(-3:3, f(-3:3), lwd = 2, lty = "dotted", col = "red")
curve(x^3, lty = "dashed", col = "blue", add = TRUE)
curve(x^4, lty = "dashed", col = "yellow", add = TRUE)
legend("bottomright",
       legend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

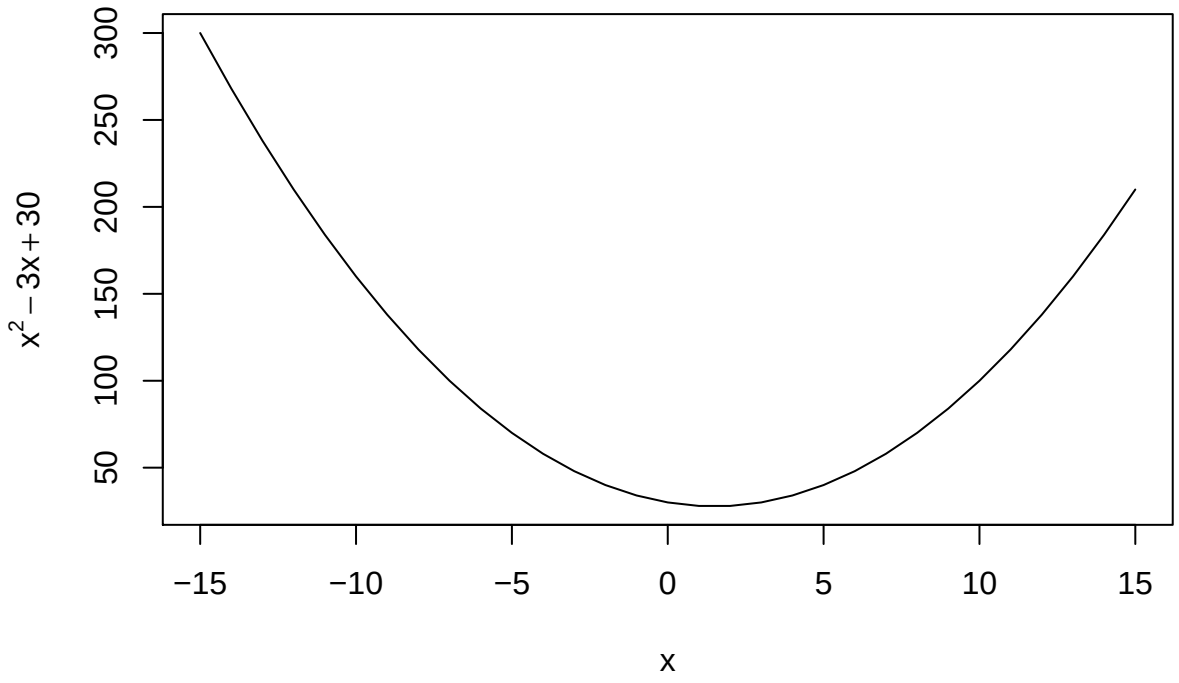


Tarea

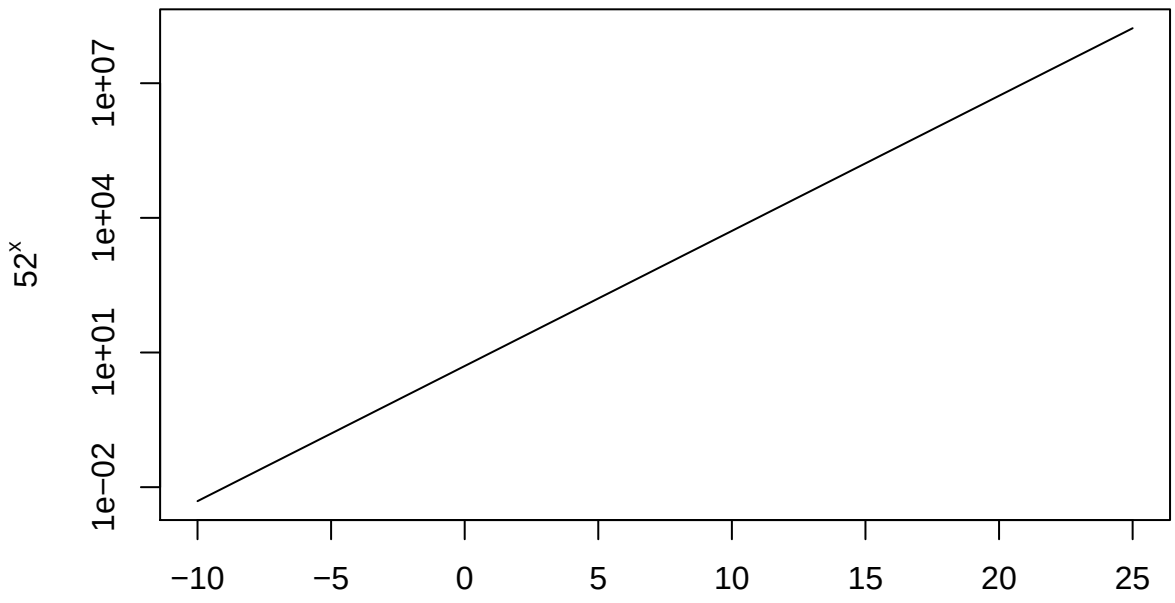
```r
# 1
curve(x^2 - 3*x + 30, -15, 15, main = "Una parábola", xlab = expression(x), ylab = expression(y = x^2 -
```



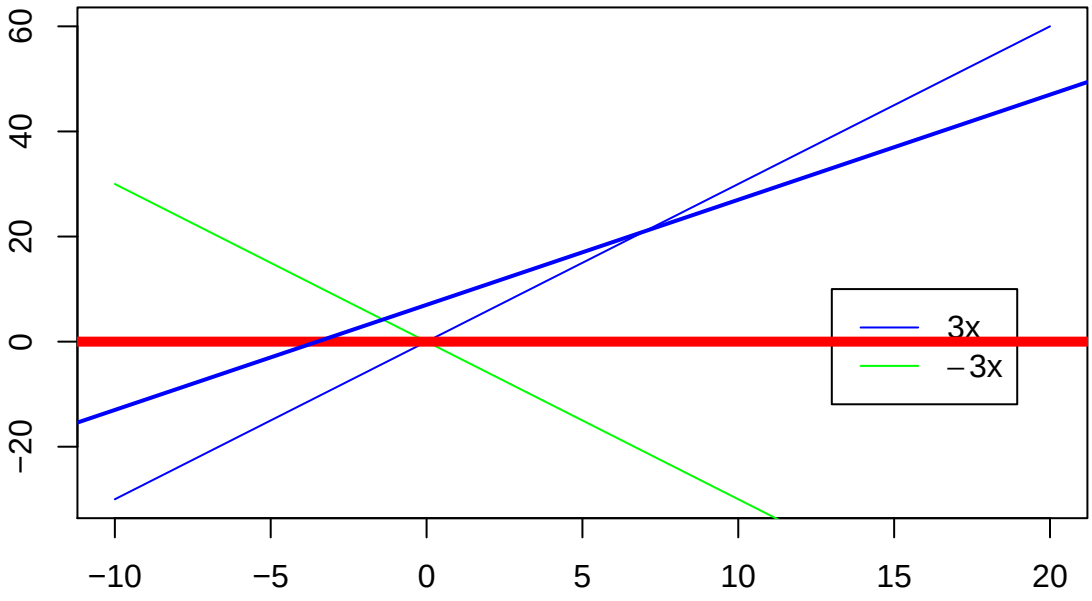

```
# 2
f = function(x) {x^2 - 3*x + 30}
i = c(-15:15)

# plot(f(i))
plot(i, f(i), type = "l", main = "Una parábola", xlab = expression(x), ylab = expression(y = x^2 - 3*x
```




```r
# 3
curve(5 * 2^x, xlim = c(-10, 25), xlab = "", ylab = expression(y = 5 * 2^x), log = "y")
```




```r
# 4
curve(3 * x, -10, 20, xlab = "", ylab = "", col = "blue", main = "2 rectas", sub = "Dos rectas con pendi
curve(-3 * x, col = "green", add = TRUE)
legend(13, 10, legend = c(expression(3*x), expression(-3*x)), col = c("blue", "green"), lwd = 1)

abline(h = 0, col = "red", lwd = 5)
abline(7, 2, col = "blue", lwd = 2)
```



Dos rectas con pendiente opuesta