Graficos en R

Victor Lopez

2023-01-13

Colores en R

Plot

```
plot(x, y)
plot(x) == plot(1:length(x), x)
```

Representar funciones

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

Parametros

```
log = "x" or "y" # Escala logaritmica en alguna coordenada

expression() # Convierte una expresion matematica en un string con formato

main("titulo") # Se puede pasar como parametro

xlab("labelx") # Se puede pasar como parametro
ylab("labely") # Se puede pasar como parametro

pch = 1 # Forma que va tener el punto, por defecto es 1

cex = 1 # Elegir tamaño del punto, por defecto es 1

col = "red" # Color del borde de los puntos

bg = "blue" # Color del relleno del punto. Solo los puntos que tengan relleno

lty = "solid" or 1 or "dashed" or 2 or "dotted" or 3 or "dotdashed" or 4 # Tipo de linea
```

```r
lwd = 1 # Line width

xlim = c(-3,3)

ylim = c(-3,3)

xaxp = c(-100, 100, 2) # De -100 a 100 y solo dos rayitas

yaxp
```

Grid de graficos

```r
par(mfrow = c(filas, columnas)) # Los graficos se iran poniendo en orden de filas

# Al final debemos devolverlo a su valor original
par(mfrow = c(1, 1))
```

Parametro type

```r
type = "p" # Puntos, valor por defecto
type = "l" # Lineas que unen puntos (No se ven)
type = "b" # Lineas que unen puntos (Si se ven). Las lineas no traspasan los puntos
type = "o" # Lo mismo que "b" pero si traspasan los puntos
type = "h" # Histograma de lineas (No es de frecuencias, necesita de x e y)
type = "s" # Histograma de escalones
type = "n" # None
```

Anadir elementos a un grafico

```r
points(x, y) # Añade a un plot, puntos de coordenadas. Puede ser vectores de puntos

abline(a, b) # Añade la recta y = bx + a
abline(v = 0) # Recta vertical en una cordenada x. Pueden ser vectores de lineas
abline(h = 0) # Recta horizontal en una cordenada y. Pueden ser vectores de lineas

lines(x, y) # x e y son los puntos, los cuales se uniran mediante lineas rectas

curve(x^2, add = TRUE) # La curva se puede especificar mediante una expresion algebraia con x
# O mediante su nombre si la hemos definido antes. Sin el add = TRUE, sa añadira aparte

segments(10, 0, 40, 0, col = "red", lwd = 4)

# length, angle y code se refieren a la forma que tendran la flecha
arrows(10, 0, 40, -10, colo = "blue", length = 0.5, angle = 5, code = 3)
```

```r
symbols(40, 0, stars = cbind(1, .5, 1, .5, 1, .5, 1, .5, 1, .5), angle = 5, add = TRUE,
        lwd = 3, inches = 0.5)

# density y angle se refieren a las lineas que van dentro del poligono formado
polygon(c(20, 30, 40), c(10, -10, 10), col = "gold", density = 3, angle = 90, lty = 4, lwd = 5)
```

Anadir texto

```r
text(1:3, 1:3, y, labels = c("A", "B", "C"), pos = 1) # x, y, labels
# pos = 1, abajo
# pos = 2, izquierda
# pos = 3, arriba
# pos = 4, derecha
# pos = 5, centro (por defecto)
```

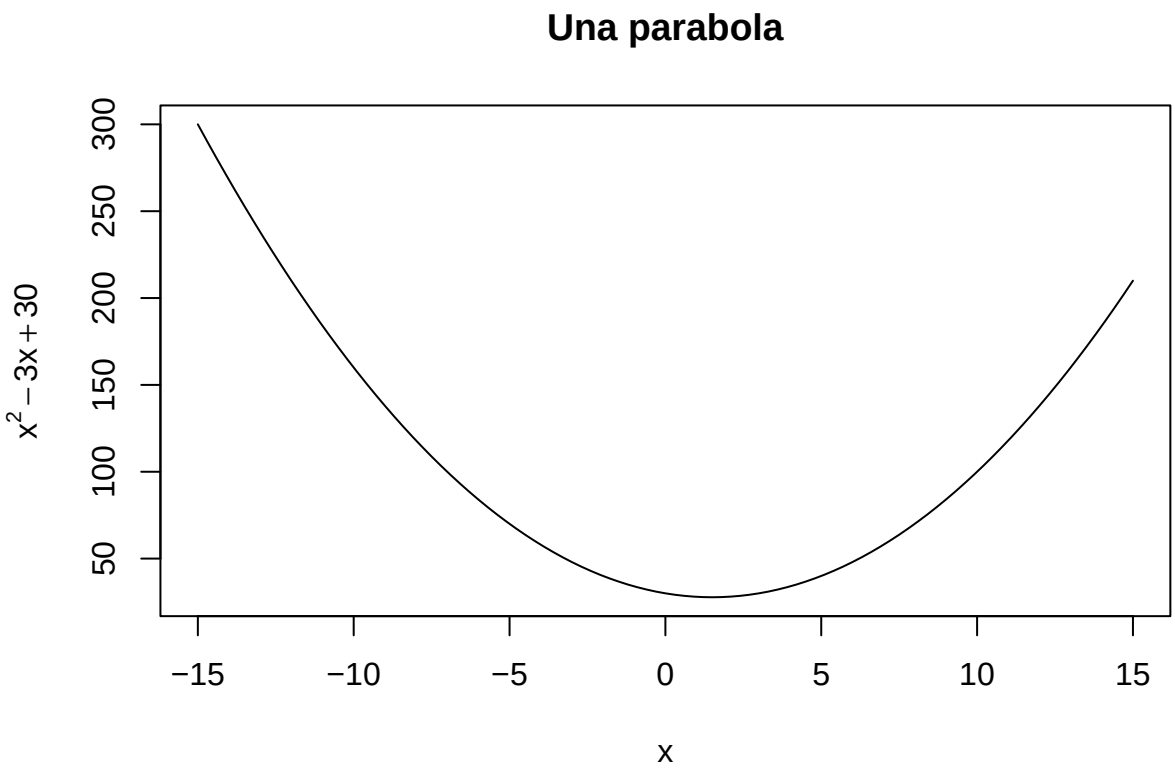
Leyendas

```r
legend("bottomright", legend = c("seno", "Coseno", "Tangente"), lwd = 2,
       col = c("red", "blue", "orangered"), lty = c("dotted", "dashed", "dashed"))
```

Ejercicios

Ejercicio 1

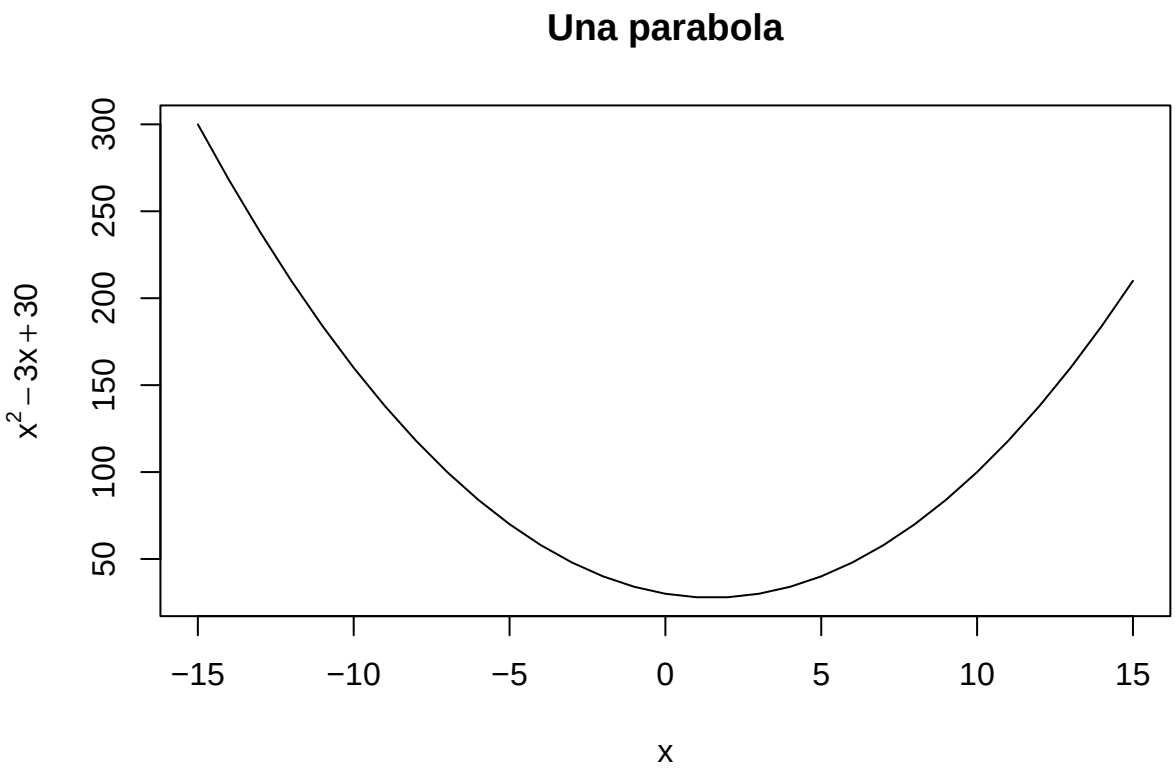
```r
curve(x^2 - 3*x + 30,-15, 15, main = "Una parabola",
      xlab = expression(x), ylab = expression(x^2 - 3*x + 30))
```



Ejercicio 2

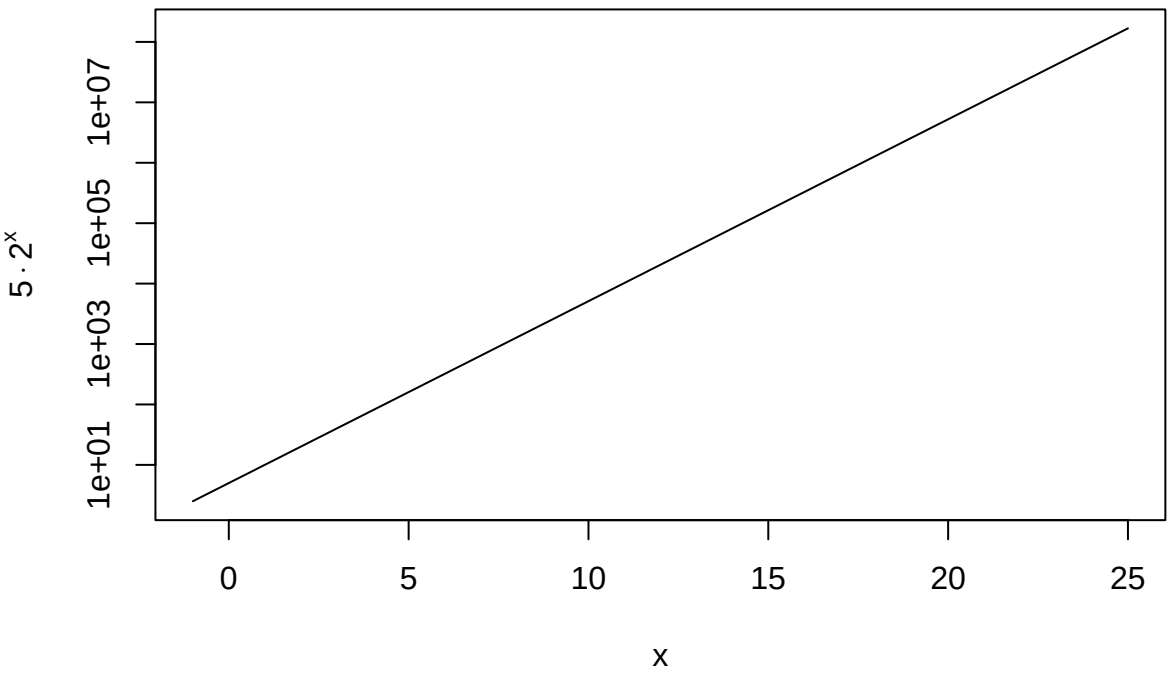
```
f = function(x) {x^2 - 3*x + 30}

plot(-15:15, f(-15:15), main = "Una parabola", type = "l",
     xlab = expression(x), ylab = expression(x^2 - 3*x + 30))
```



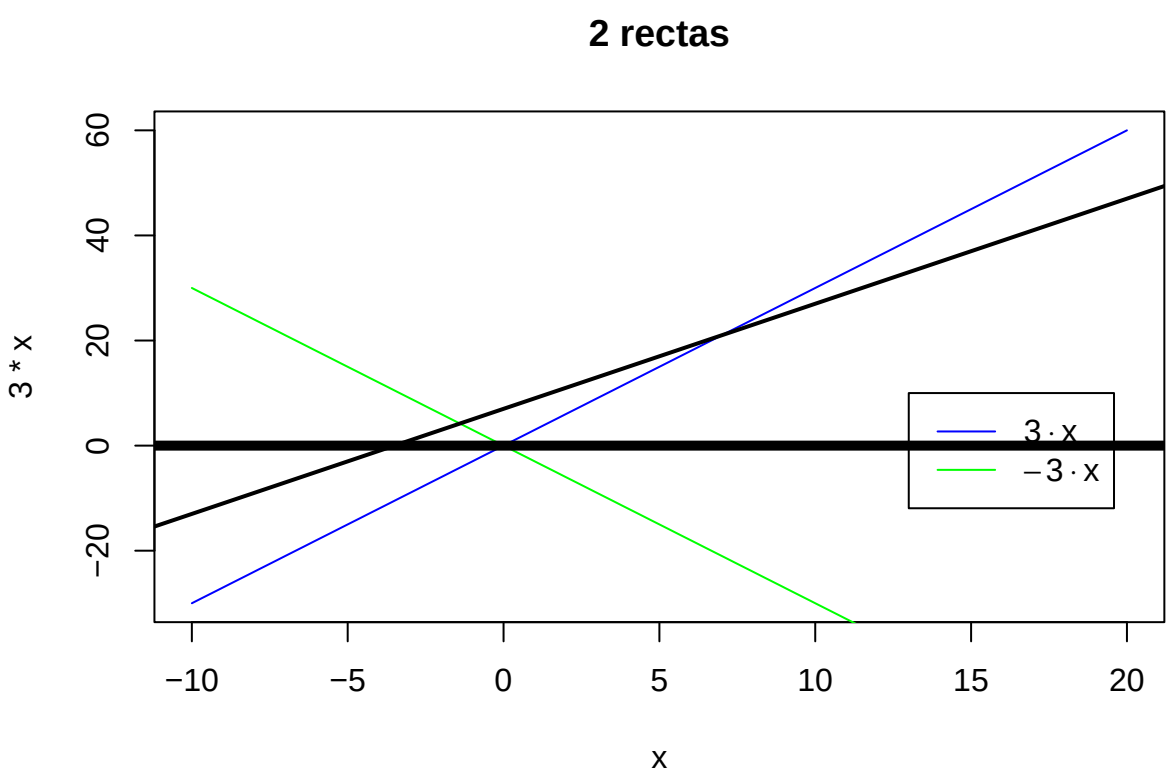
Ejercicio 3

```
curve(5 * 2^x, -1, 25, ylab = expression(y = 5 %.% 2^x, -1, 25), log = "y" )
```



Ejercicio 4

```
curve(3 * x, -10, 20, col = "blue", main = "2 rectas",
      sub = "Dos rectas con pendiente opuesta")
curve(-3 * x, add = TRUE, col = "green")
legend(13, 10, legend = c(expression(3 %.% x), expression(-3 %.% x)), col = c("blue", "green"),
       lwd = 1, lty = c(1, 1))
abline(h = 0, lwd = 5)
abline(7, 2, lwd = 2)
```



Dos rectas con pendiente opuesta