

Tarea 6- Introduccion a la Representacion Grafica

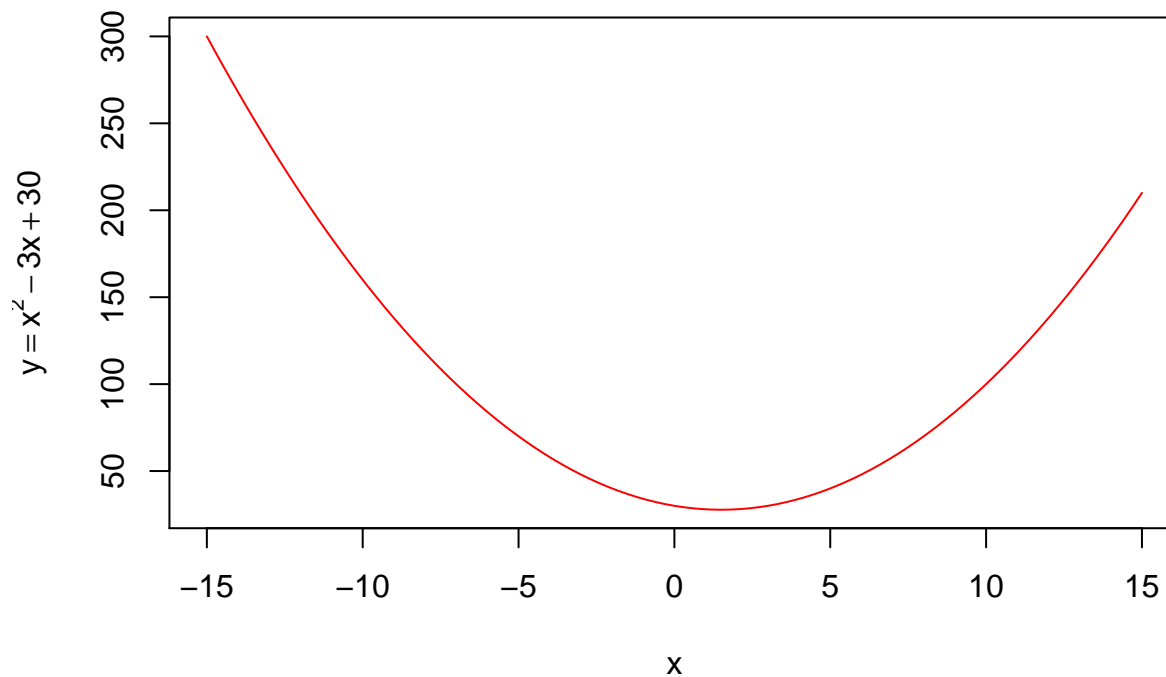
Brigido Vicuna

8/2/2020

Con una sola instrucción, dibujad el gráfico de la función $y = x^2 - 3x + 30$ entre -15 y 15. De titulo, poned “Una Parabola”. De etiquetas, en el eje 0X poned, en formato matemático, “x”; y en el eje 0Y, introducid $y = x^2 - 3x + 30$ también en formato matemático. Tenéis que utilizar la función `curve()`.

```
plot(c(-15:15),(c(-15:15)^2-3*c(-15:15)+30),  
     type = "n",  
     main = "Una parabola",  
     xlab = expression(x),  
     ylab = (expression(y==x^2-3*x+30)))  
curve(x^2-3*x+30, add = TRUE, col = "red")
```

Una parabola

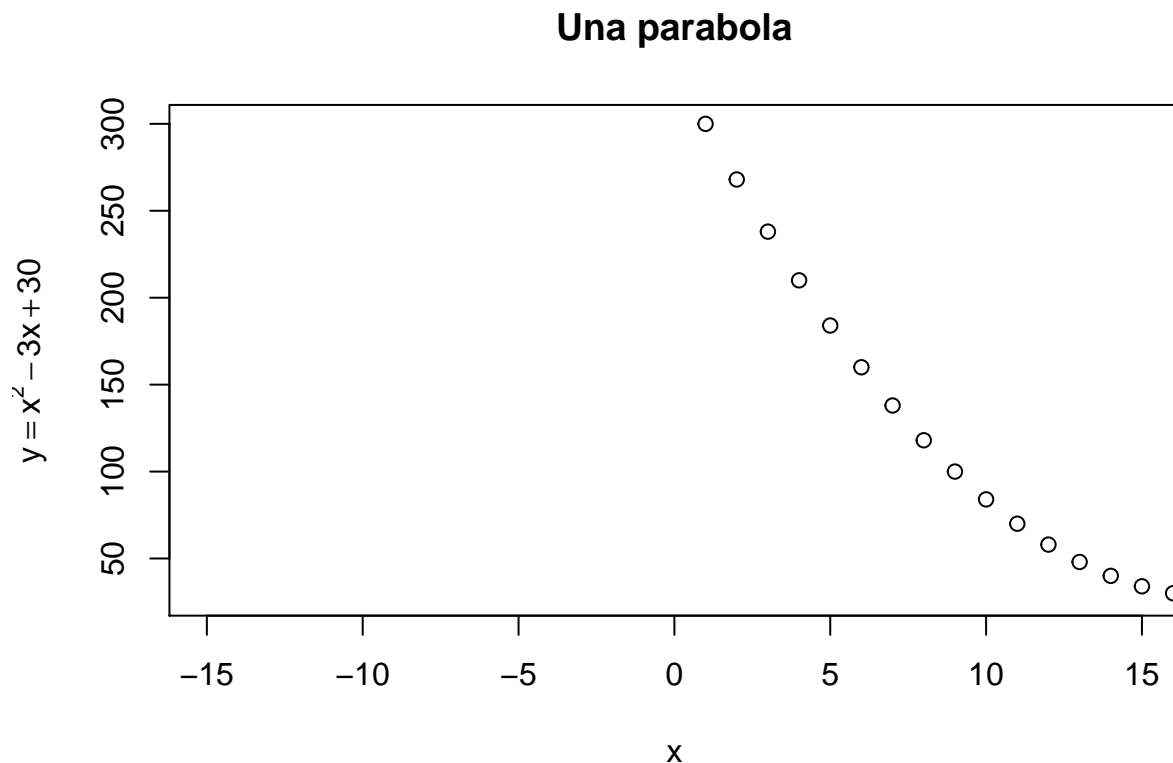


Considerando lo que habéis obtenido en el ejercicio anterior y siendo $y = f(x) = x^2 - 3 * x + 30 \in I = [-15 : 15]$, si en vez de utilizar la función `curve()`, utilizamos la función `plot()`, ¿es correcta la sentencia `plot(f(I))` para representar la curva f en el intervalo I? En otras palabras, dan ambas sentencias la misma

gráfica? Obviamente, en la sentencia `plot(f(I))` se han omitido el resto de parámetros requeridos en el ejercicio anterior porque no influyen para nada en la curva. Tanto si la respuesta es afirmativa como negativa, cread la función `f` en R y argumentad vuestra respuesta, considerando todos los parámetros requeridos (título y etiquetas de ambos ejes).

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

plot(y(I),
     main = "Una parabola",
     xlab = expression(x),
     ylab = (expression(y==x^2-3*x+30)),
     xlim = c(-15,15))
```



No es correcta esta sentencia para representar `f(I)`. La función `f(I)` es evaluada en función de índices, mientras que en el ejercicio anterior, la función se evalúa en función del rango desde -15 a 15.

Comparando ambas expresiones en el rango de -15 a 15 se tiene:

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

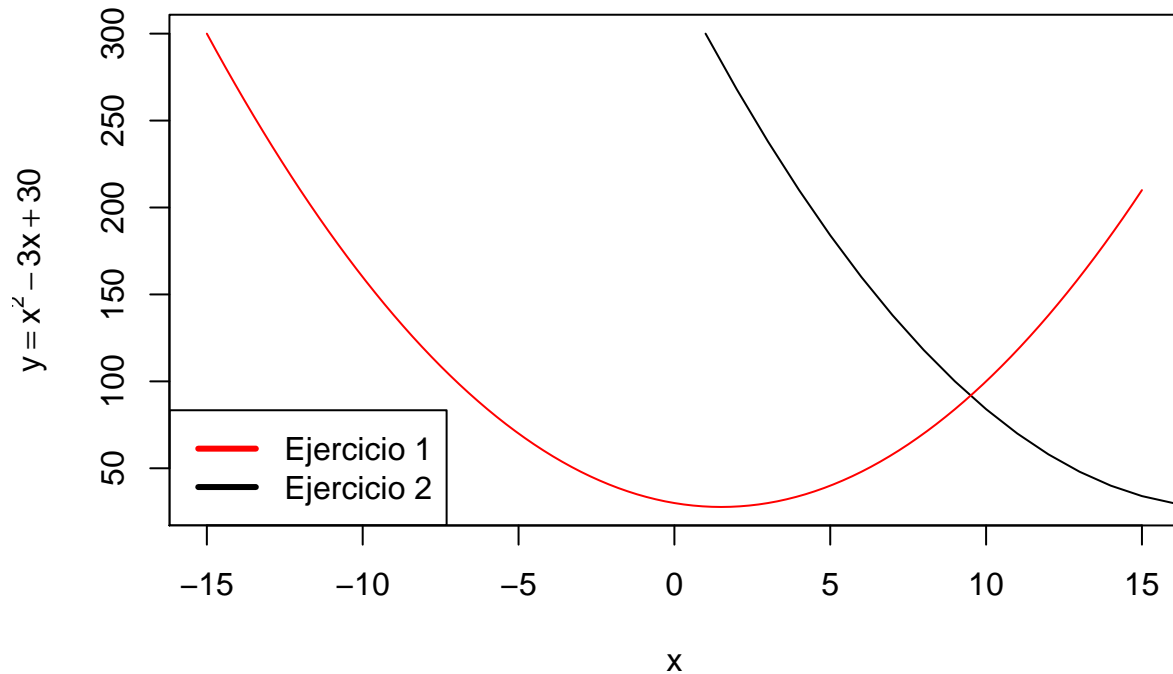
plot(y(I),
     main = "Una parabola",
     xlab = expression(x),
     ylab = (expression(y==x^2-3*x+30)),
```

```

xlim = c(-15,15),
type = "l")
curve(x^2-3*x+30, from = -15, to = 15, add = TRUE, col = "red")
legend("bottomleft",col=c("red","black"),legend=c("Ejercicio 1","Ejercicio 2"),
lwd=3, bty="l")

```

Una parabola



Dibuja un gráfico semilogarítmico de la función $y = 5 \cdot 2^x$ entre -10 y 25 . Utilizad la función `curve()`. Mostrad solo la etiqueta del eje OY, que ponga " $y = 5 \cdot 2^x$ " en formato matemático.

```

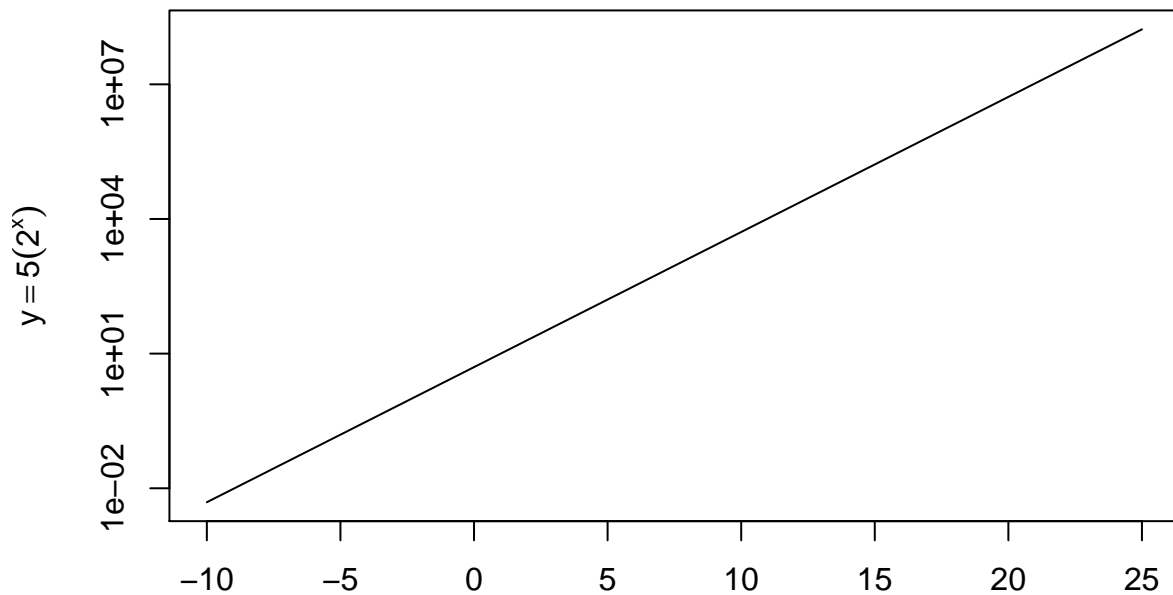
y = function(x){
  5*2^x
}
x = c(-10:25)
plot(x,y(x),
type = "n",
ylab = expression(y==5*(2^x)),
xlab = "",
log = "y")
curve(5*2^x, from = -10, to = 25, add = TRUE)

```

```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "form" is not a graphical
## parameter

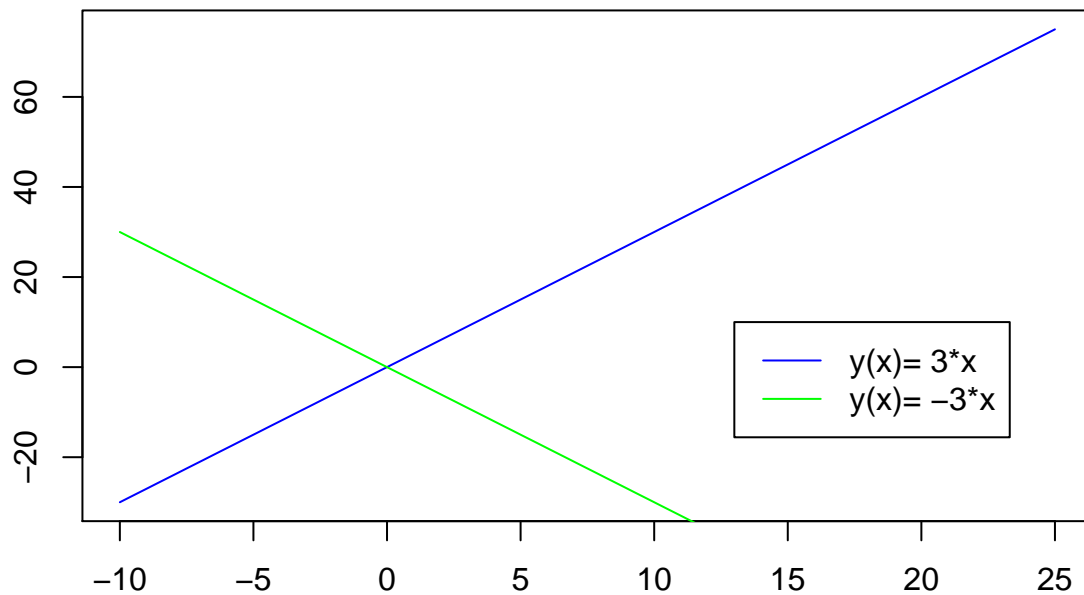
```



Dibuja el gráfico de la función $y_1 = 3x$ utilizando la función `curve()`. Añade la curva $y_2 = -3x$, entre -10 y 20. El gráfico no debe mostrar ninguna etiqueta. La primera curva debe ser de color azul y la segunda, de color verde. Ponedle de título “2 rectas” y de subtítulo “Dos rectas con pendiente opuesto”. Añadid al gráfico un recuadro (con la esquina superior izquierda en el punto (13,10)) que indique que la función $3x$ es la azul y la $-3x$ verde.

```
y_1 = function(x){
  3*x
}
y_2 = function(x){
  -3*x
}
x = (c(-10:20))
}
plot(x,y_1(x),
     type = "n",
     xlim = c(-10,25),
     xlab = "",
     ylab = "",
     main = "2 rectas",
     sub = "Dos rectas con pendiente opuesto")
legend(x = 13, y = 10,col=c("blue","green"), legend=c("y(x)= 3*x","y(x)= -3*x"),
      lwd=1, bty="l")
curve(3*x, add = TRUE, col = "blue")
curve(-3*x, add = TRUE, col = "green")
```

2 rectas

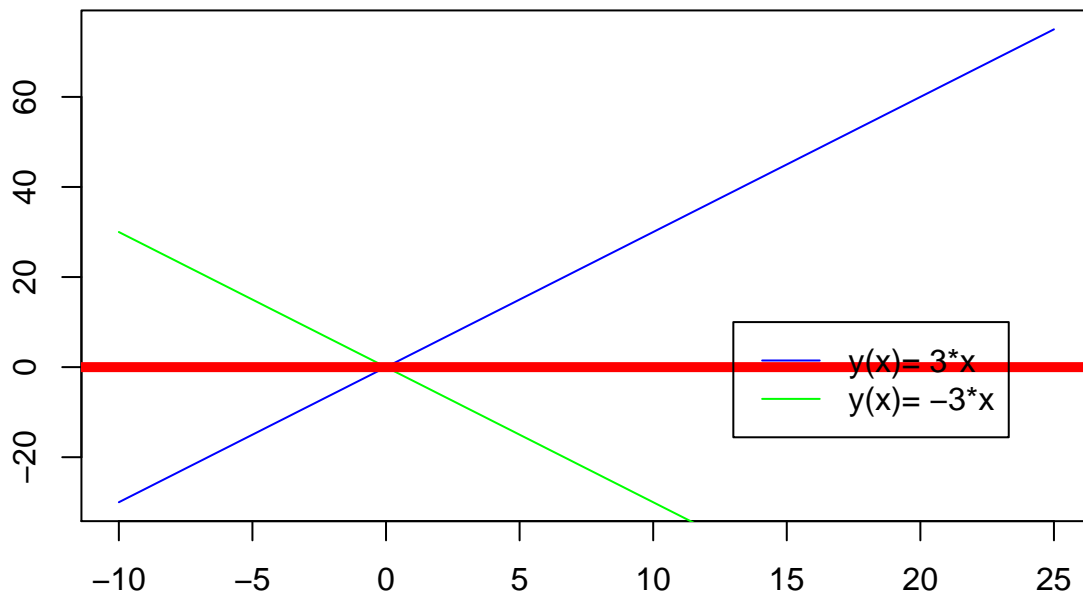


Dos rectas con pendiente opuesto

Dad la instrucción que añada a un gráfico anterior la recta horizontal $y = 0$ de color rojo con un grosor de 5 puntos.

```
y_1 = function(x){
  3*x
}
y_2 = function(x){
  -3*x
}
x = (c(-10:20))
}
plot(x,y_1(x),
     type = "n",
     xlim = c(-10,25),
     xlab = "",
     ylab = "",
     main = "2 rectas",
     sub = "Dos rectas con pendiente opuesto")
curve(3*x, add = TRUE, col = "blue")
curve(-3*x, add = TRUE, col = "green")
abline(h = 0, lwd = 5, col = "red")
legend(x = 13, y = 10,col=c("blue","green"), legend=c("y(x)= 3*x","y(x)= -3*x"),
      lwd=1, bty="l")
```

2 rectas

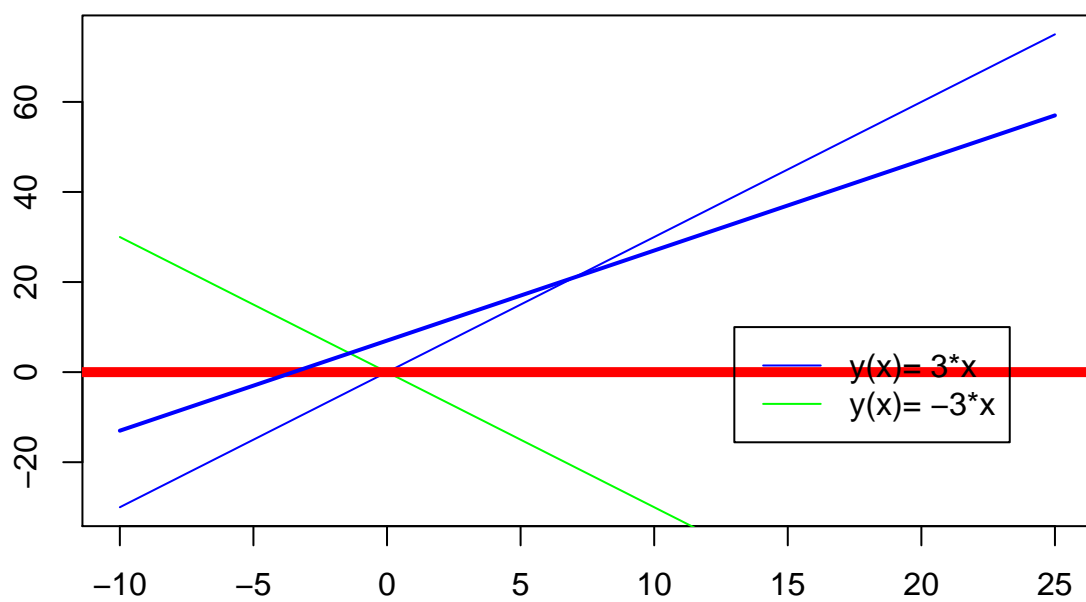


Dos rectas con pendiente opuesto

Dad la instrucción que añada a un gráfico anterior la recta $y = 2x+7$ de color azul con un grosor de 2 puntos.

```
y_1 = function(x){
  3*x
}
y_2 = function(x){
  -3*x
}
x = (c(-10:20))
}
plot(x,y_1(x),
      type = "n",
      xlim = c(-10,25),
      xlab = "",
      ylab = "",
      main = "2 rectas",
      sub = "Dos rectas con pendiente opuesto")
curve(3*x, add = TRUE, col = "blue")
curve(-3*x, add = TRUE, col = "green")
abline(h = 0, lwd = 5, col = "red")
lines(x,2*x+7,col="blue",lwd=2)
legend(x = 13, y = 10,col=c("blue","green"), legend=c("y(x)= 3*x","y(x)= -3*x"),
       lwd=1, bty="l")
```

2 rectas



Dos rectas con pendiente opuesto