

Graficos-practice

Francisco

2024-10-17

Este rmarkdown sirve para practicar las funciones del tema de representación gráfica con `plot()` del curso de Estadística descriptiva con R y Python

Gráficos con la función `plot`

Así se hace un gráfico de puntos con la función `plot()`:

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

Si solo suministramos `x` sin el parámetro `y` a la función `plot(x,y)`, R tomará a `x` como `y`, y a `x` como `1:length(x)`, o bien: `plot(x) = plot(1:length(x),x)`

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

```
f <- function(x){x^2}
plot(f,col="red")
```

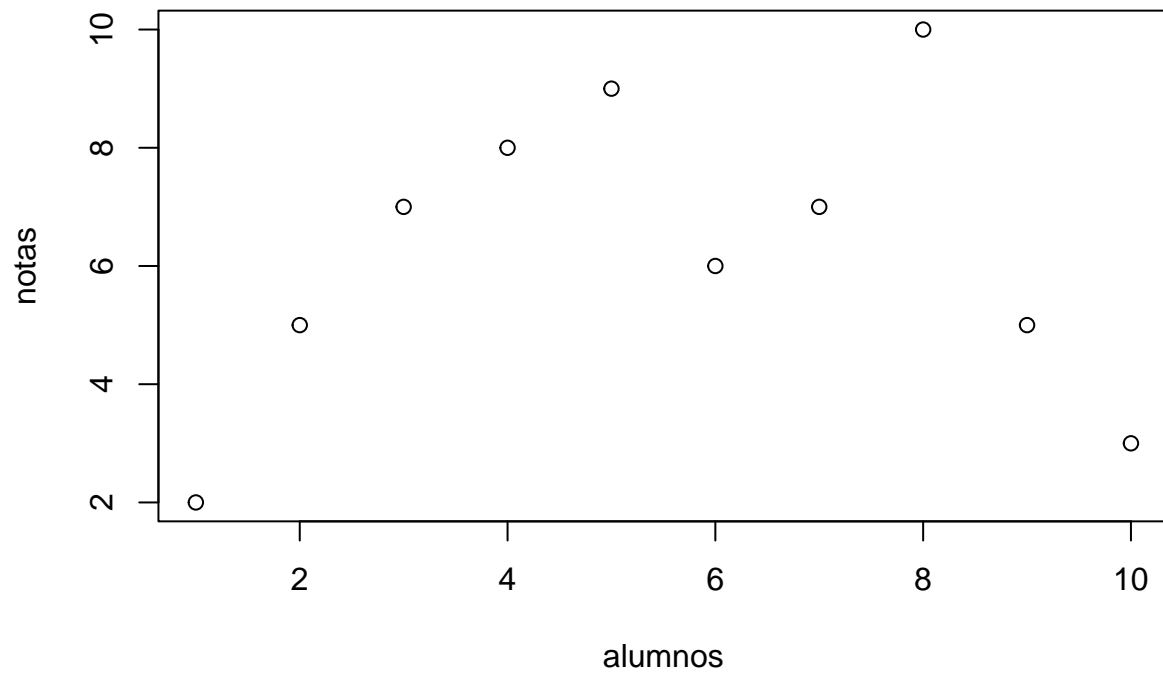
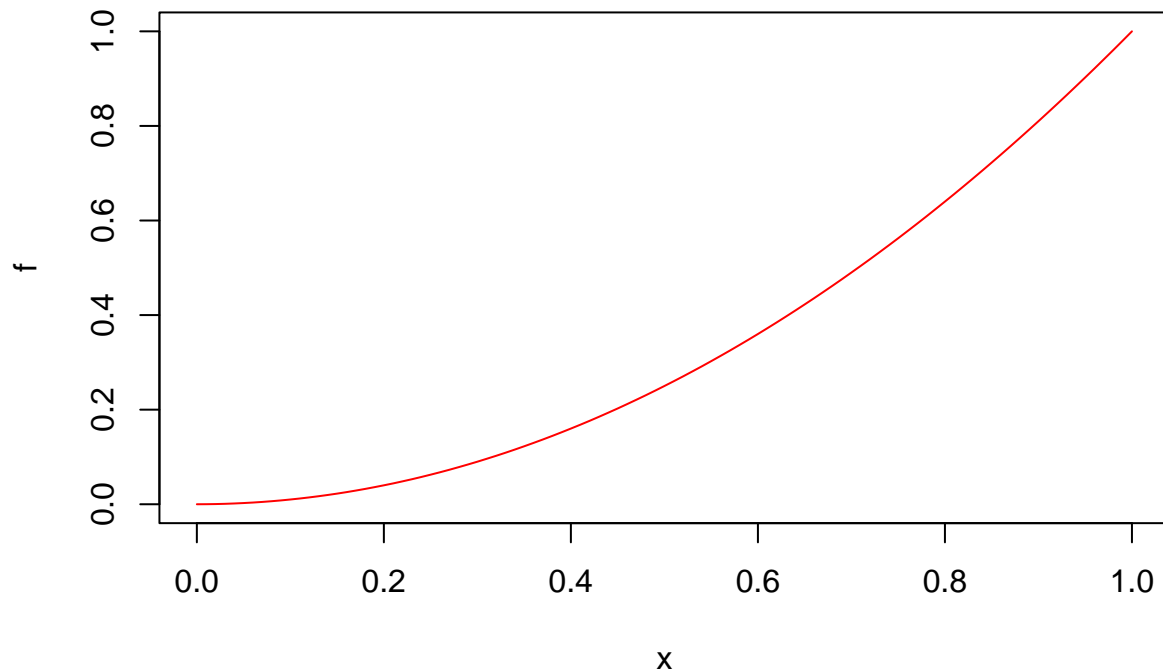


Figure 1: Fig.1 Gráfico con plot



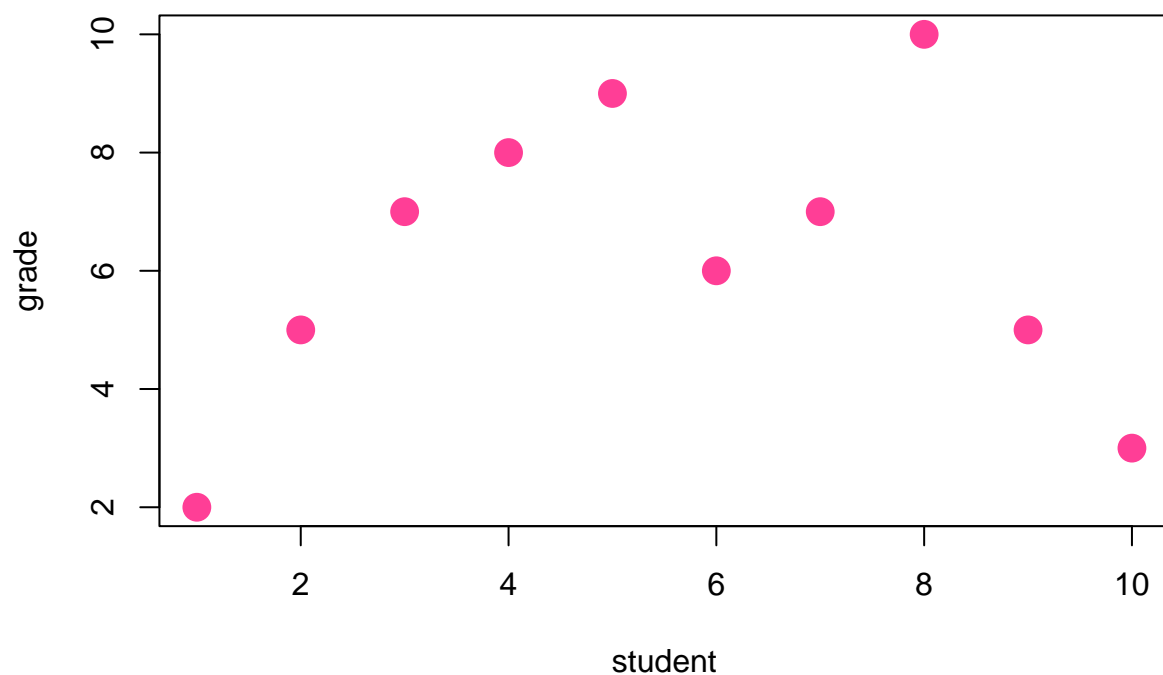
Parámetros de la función plot

Modificamos el gráfico de la Fig. 1 con los siguientes parámetros:

- `main('título')`: para indicar un título al gráfico. También podemos colocar una ecuación con LaTeX con `expression()`
- `xlab()`, `ylab()`: Títulos a los ejes
- `pch = n`: Todos los códigos de los símbolos se encuentran en las diapositivas
- `col = "color name"`: los códigos de colores se encuentran en el repositorio
- `log`: indicamos que queremos el gráfico en escala logarítmica

```
plot(alumnos,notas,main= "Notas de 10 estudiantes",xlab = "student",ylab = "grade",col = "violetred1",p
```

Notas de 10 estudiantes



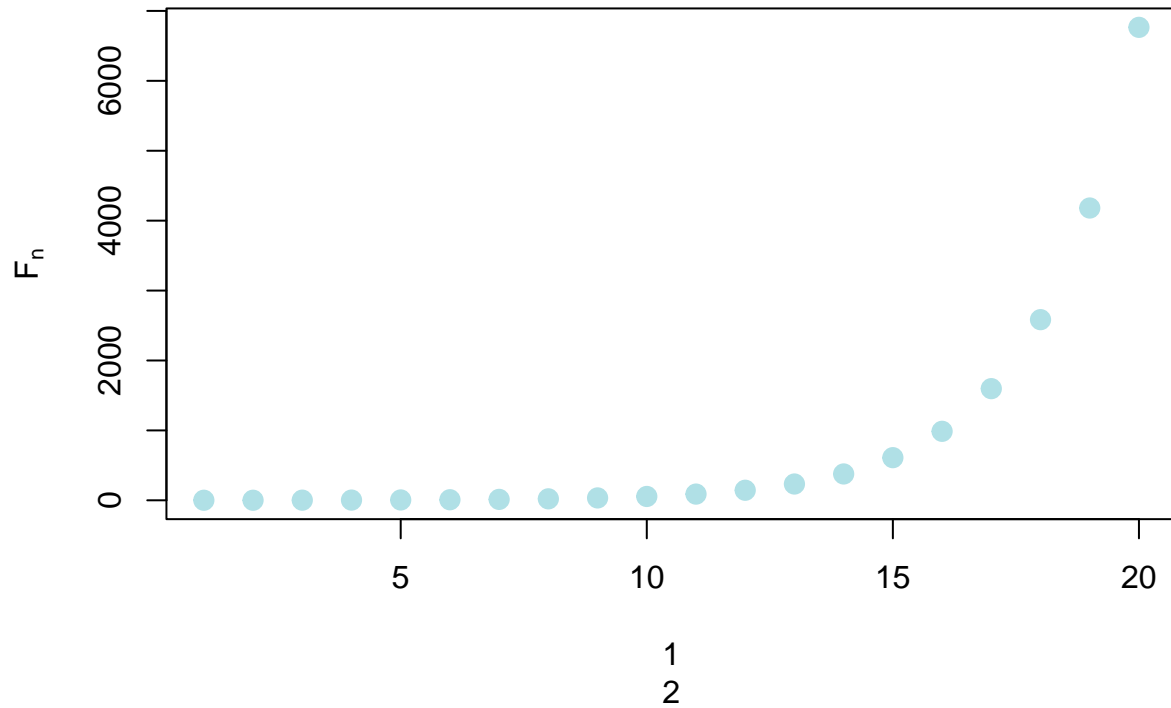
Graficando la sucesión de Fibonacci.

```
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
```

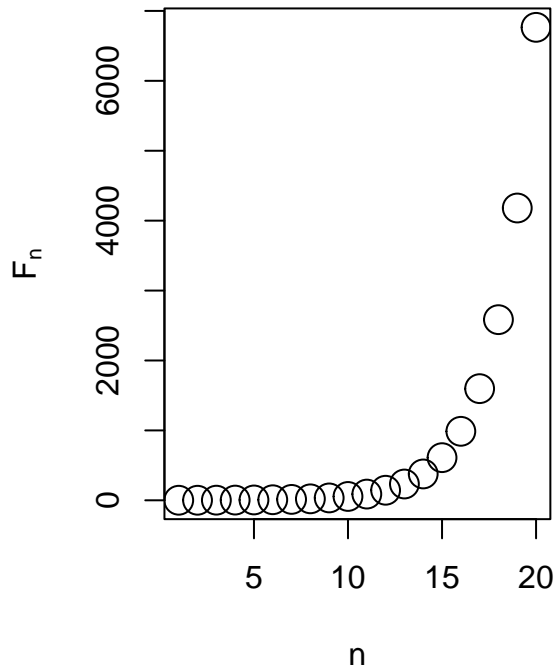
```
plot(fib,xlab = n,ylab = expression(F[n]),main = "Sucesión de Fibonacci",pch = 20,col="powderblue",cex=2)
```

Sucesión de Fibonacci

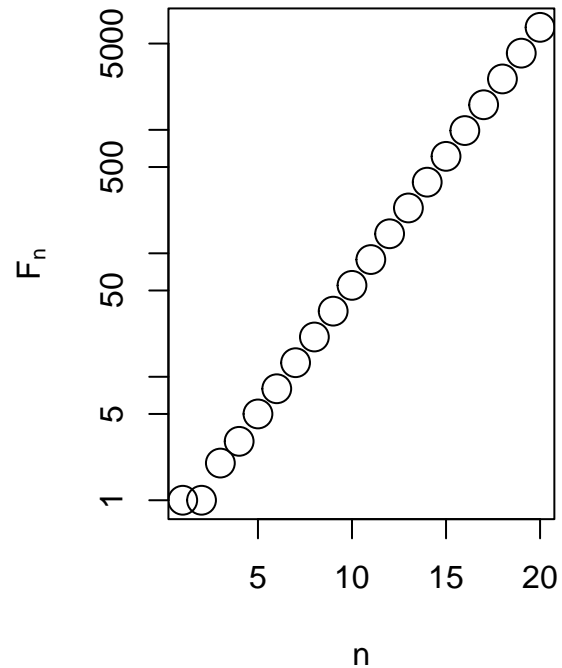


Si queremos cambiar la distribución de los gráficos, podemos usar el parámetro de configuración `par(mfrow=c(...))` indicando la disposición. En el ejemplo siguientes se utiliza la disposición 1 fila, 2 columnas.

Sucesión de Fibonacci



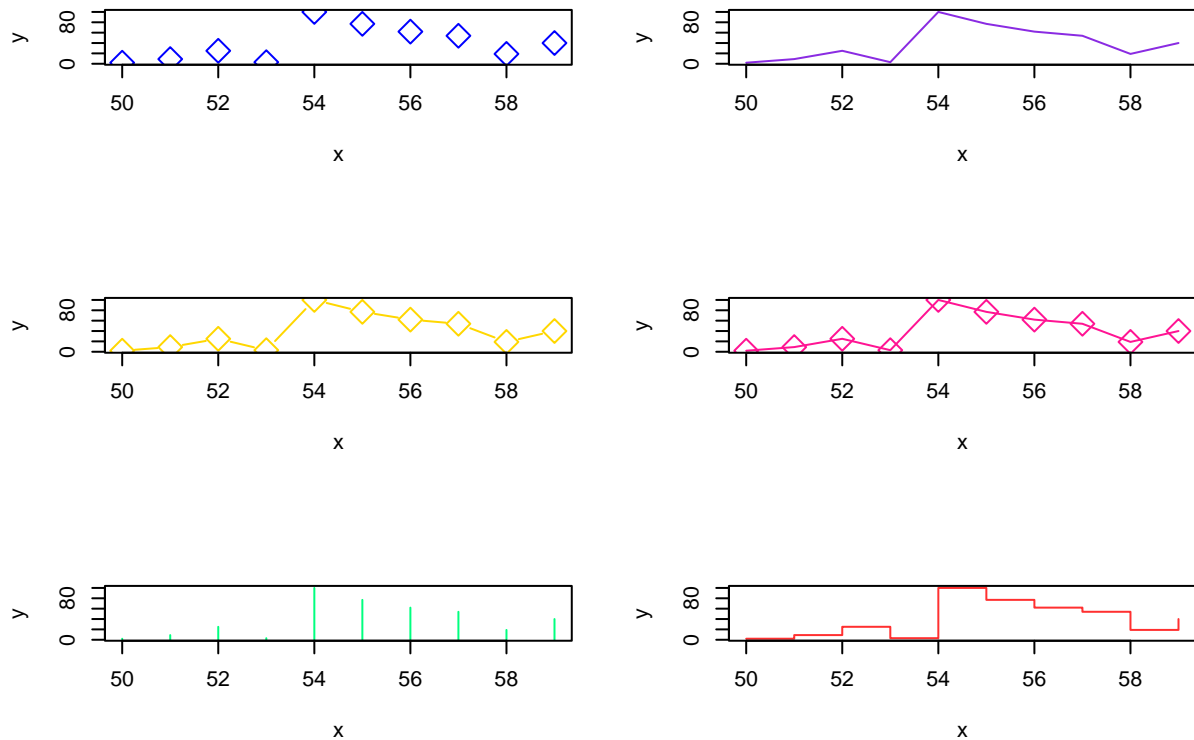
Sucesión de Fibonacci



Parámetros de la función plot - tipo de gráfico

- 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

```
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))
```

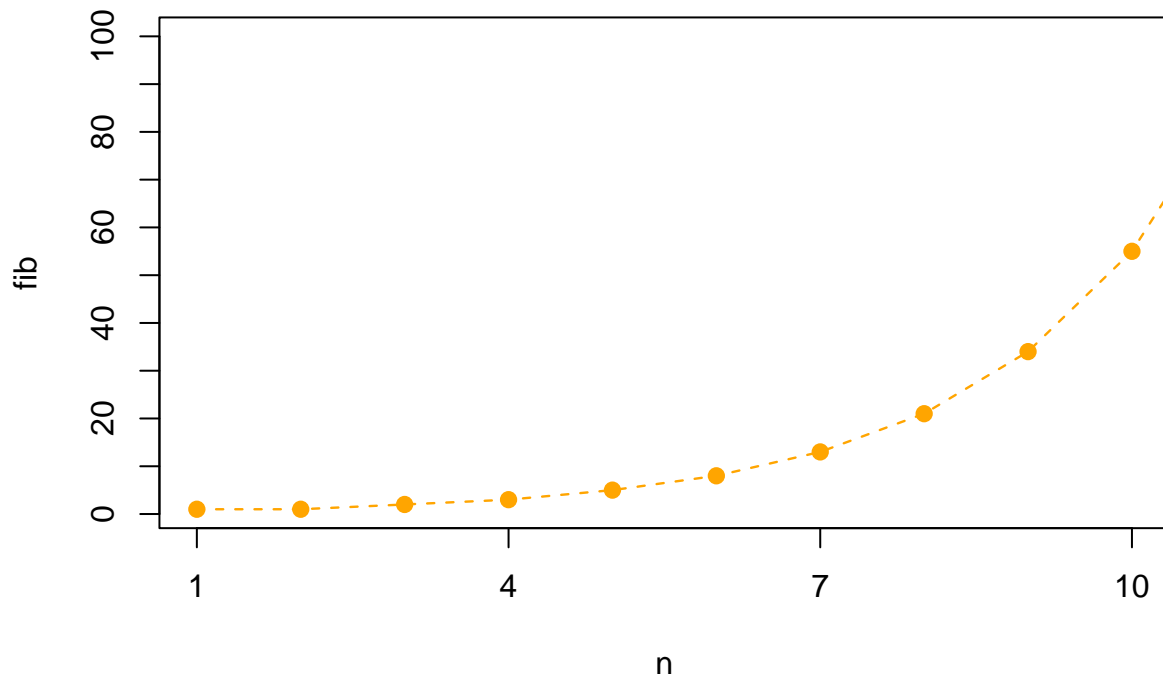
Más parámetros de la función plot

- `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

El gráfico siguiente muestra el uso de los parámetros anteriores:

```
plot(n,fib,main="Función Fibonacci",type = "o",cex=1.2,pch=16,
     col = 'orange',lty="dashed",lwd = 1.2,xlim = c(1,10),ylim =c(1,100),
     yaxp = c(1,10,3),yaxp = c(0,100,10))
```

Función Fibonacci



Añadiendo elementos al gráfico

- `points(x,y)`: añade un punto de coordenada (x,y) en un gráfico ya existente

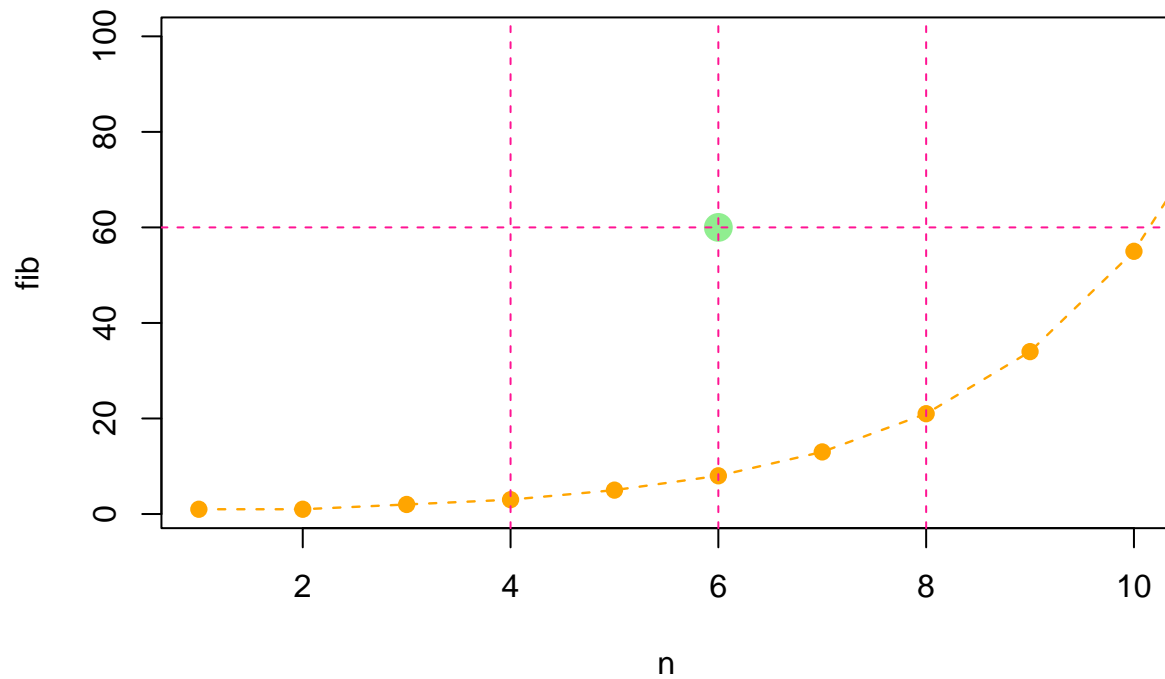
`abline()`: - `abline(x,y)`: dibuja una recta de la forma $y = bx + a$

- `abline(h = x_0)`: dibuja una línea recta horizontal en h (x_0 puede ser vector numérico)
- `abline(v = y_0)`: dibuja una línea vertical (x_0 puede ser vector de números)

```
plot(n,fib,main="Función Fibonacci",type = "o",cex=1.2,pch=16,
     col = 'orange',lty="dashed",lwd = 1.2,xlim=c(1,10),ylim=c(1,100))

points(6,60,col= "lightgreen",cex=2, pch = 16) #añadimos un punt(6,60)
abline(h=60,v = c(4,6,8),lty=2,col="deeppink") #añadimos rectas
```


Función Fibonacci

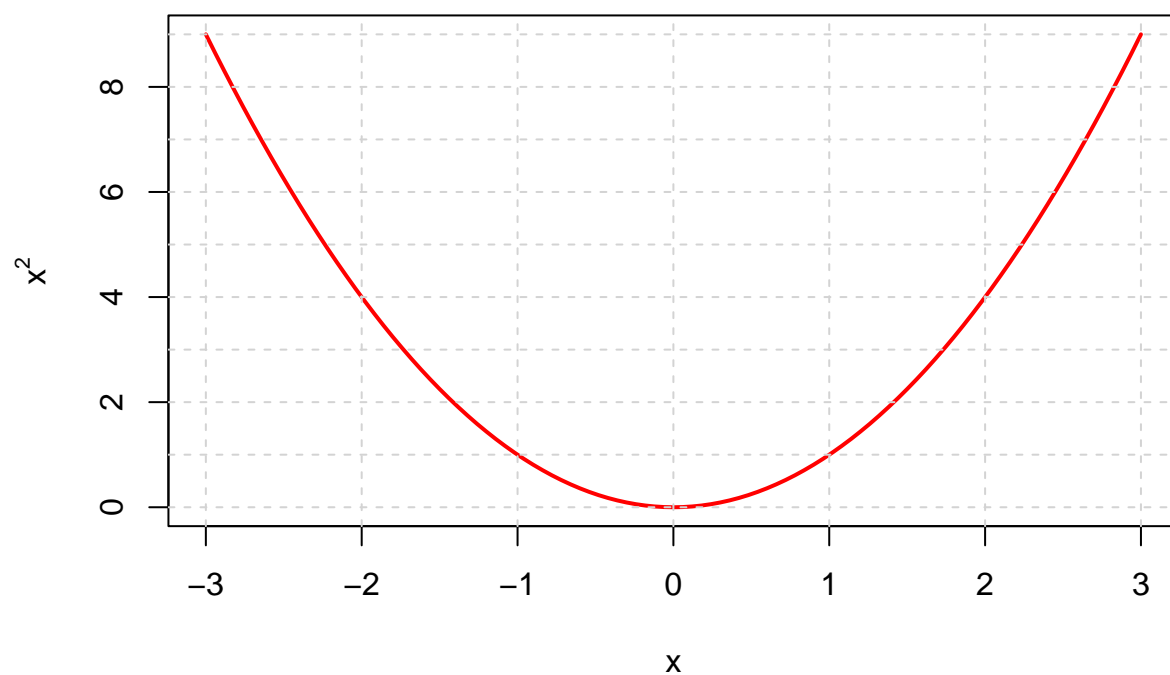


Añadiendo rectas a la gráfica de una función de una parábola $f(x) = x^2$

```
f <- function(x){x^2}

plot(f, xlim = c(-3,3), main = "Función parábola",
     xlab = "x", ylab = expression(x^2), col = "red", lwd=2)
abline(h = 0:9, v = -3:3, col = "lightgrey", lty=2)
```

Función parábola



Añadiendo rectas a la gráfica de una función tangente $f(x) = \tan(x)$

```
plot(tan, xlim = c(-pi,pi), ylim = c(-4,4), main = "Función parábola",  
      xlab = "x", ylab = expression(tan(x)), col = "lightgreen", lwd=2)  
abline(v = c(-pi/2, pi/2), col = "red")
```

Función parábola

