

Tema 5: Representación gráfica

Daniel Eduardo Macias Estrada

17/4/2021

Gráficos con la función plot

- `plot(x,y)`: para dibujar un gráfico básico de puntos siendo x y vectores numéricos
 - `plot(x) = plot(1:length(x),x)`
- `plot(x, función)`: para dibujar el gráfico de una función

```
alumnos = c(1:10)
notas = c(6,4,8,9,10,2,5,1,8,9)
plot(alumnos, notas)
```

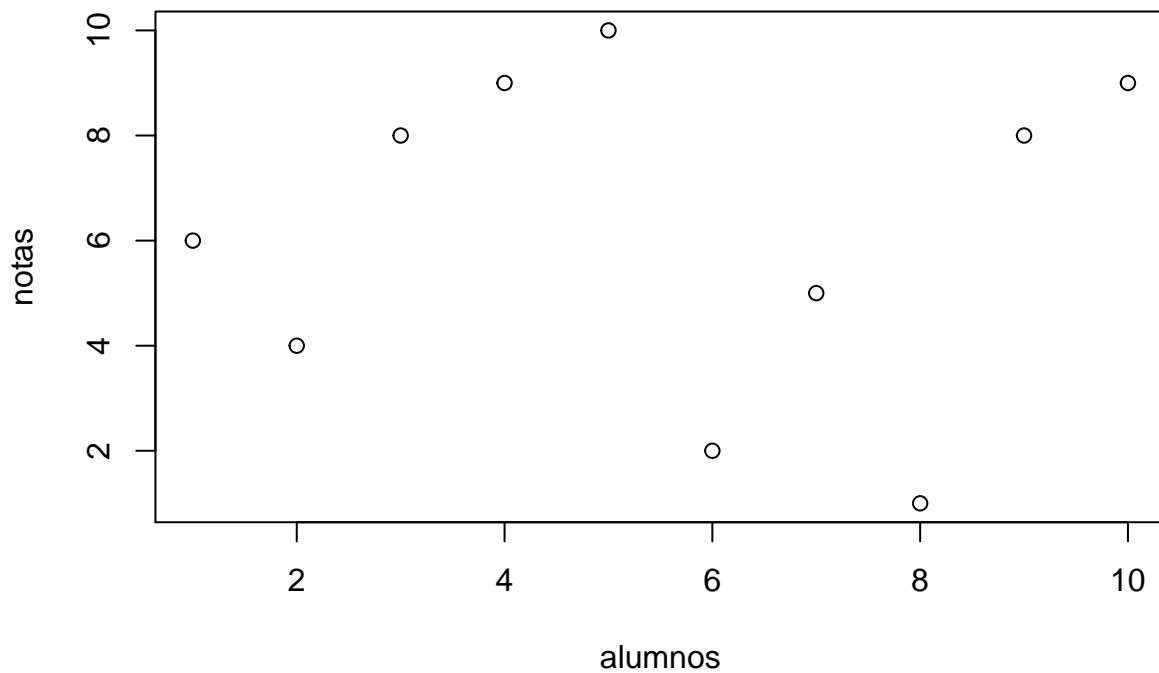
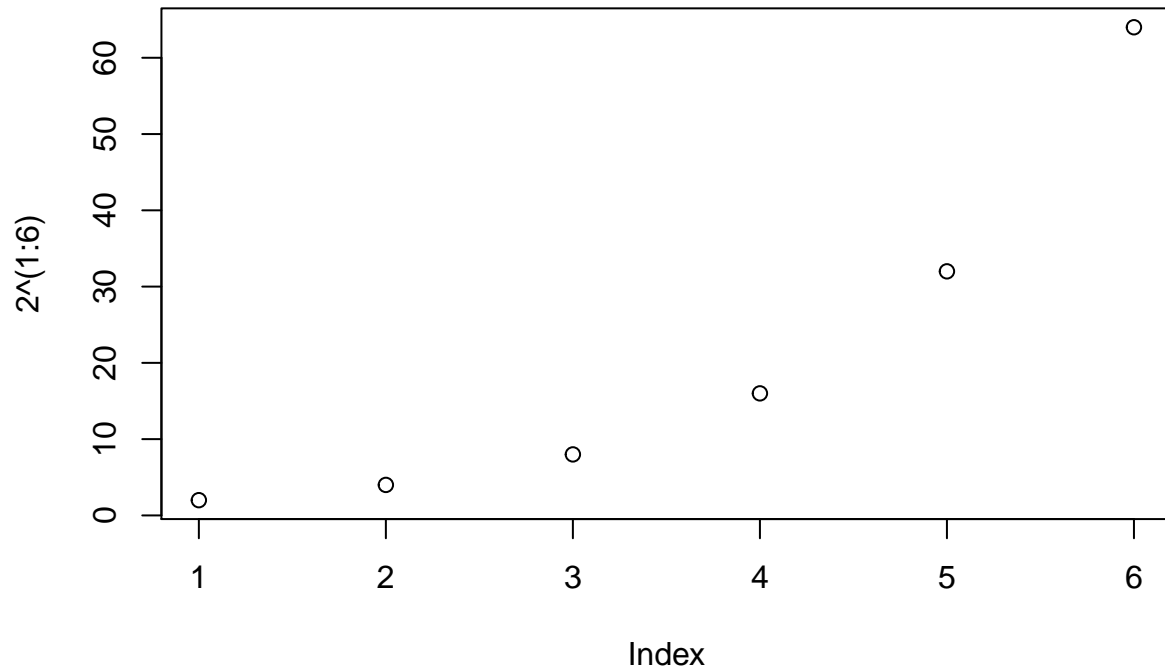


Figure 1: Gráfica de las calificaciones de 10 alumnos

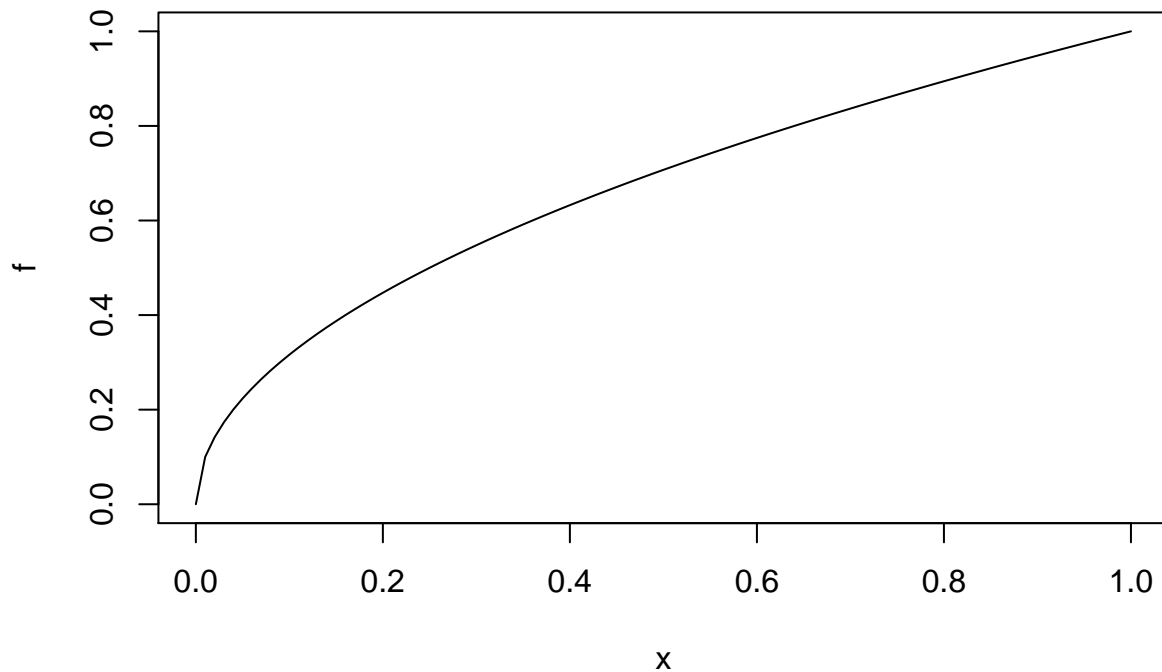
Si no incorporamos vector `y`, R nos va a tomar el parámetro `x` como si fuese el vector de datos `y` : `plot(1:n, x)`

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



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

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



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 queráis poner una expresión matemática, tenéis 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 $n = 0, 1, \dots, 25$. 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.

Ejemplo

```
par(mfrow = c(1,2))
plot = plot(exp(1:20), xlab = "Indice", ylab = expression(e^{1:20}),
            main = "Escala lineal")
plotLog = plot(exp(1:20), log = "y", xlab = "Indice",
              ylab = expression(e^{1:20}),
              main = "Escala logaritmica en el eje y")
```

```
n = 1:20
f = 1/sqrt(5) * ((1+sqrt(5))/2)^n - 1/sqrt(5) * ((1 - sqrt(5))/2)^n
```

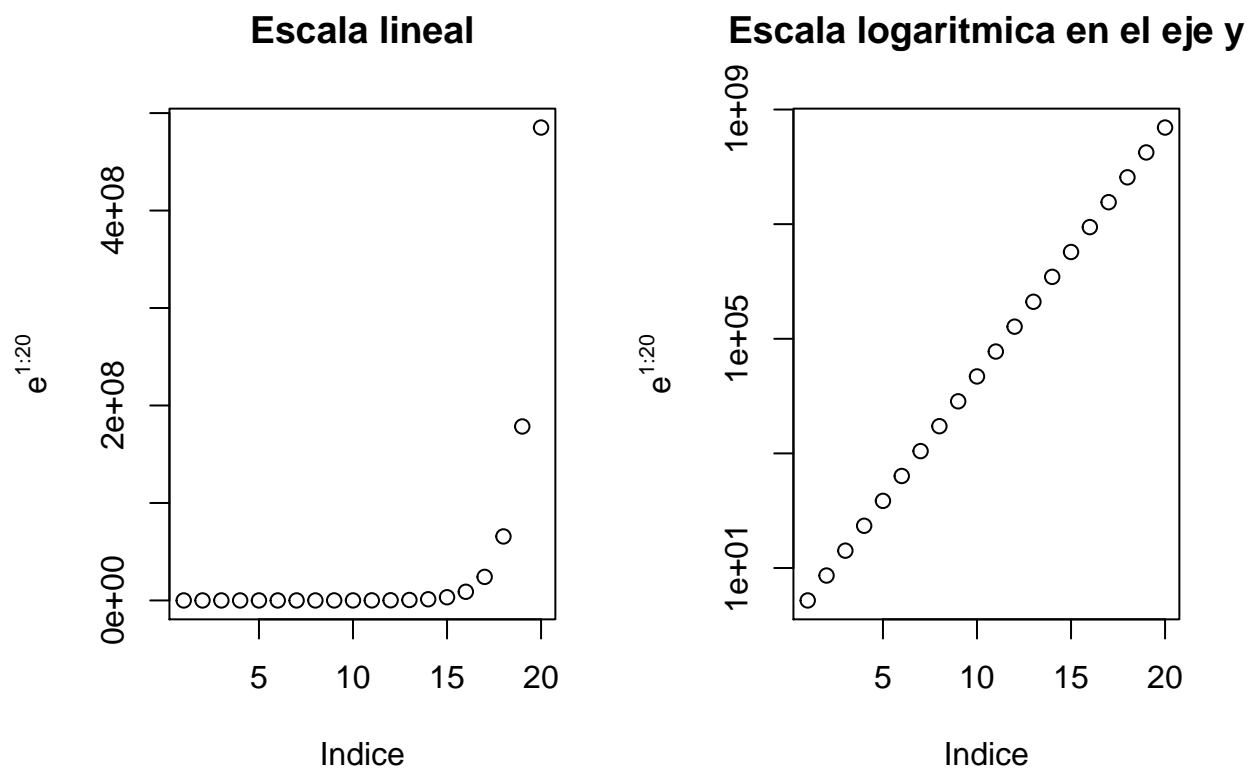
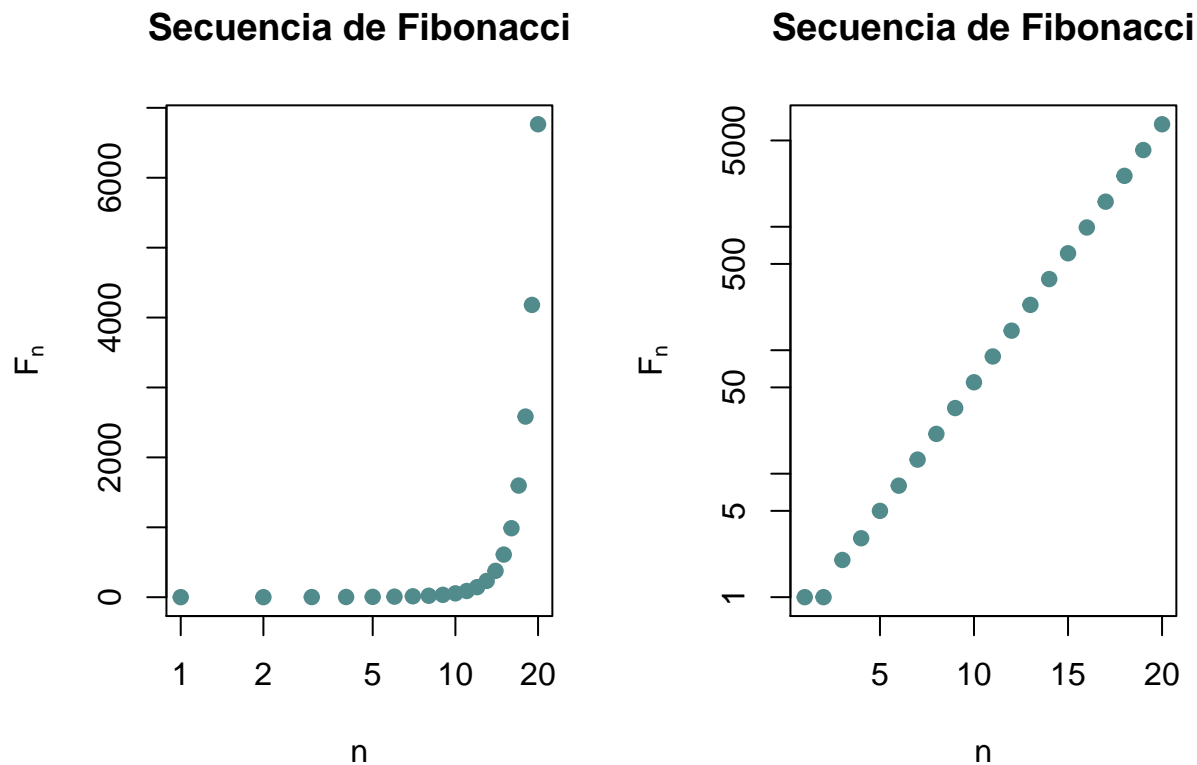


Figure 2: Primeros parámetros a considerar

```
par(mfrow = c(1,2))
plot(f, xlab = "n", ylab = expression(F[n]),
     main = "Secuencia de Fibonacci", pch = 20,
     cex = 1.5, col = "darkslategray4", log = "x")
plot(f, xlab = "n", ylab = expression(F[n]),
     main = "Secuencia de Fibonacci", pch = 20,
     cex = 1.5, col = "darkslategray4", log = "y")
```

Ejemplo Fibonacci



Otros parámetros son los siguientes

- **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 símbolos). 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

Ejemplos

```

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")

```

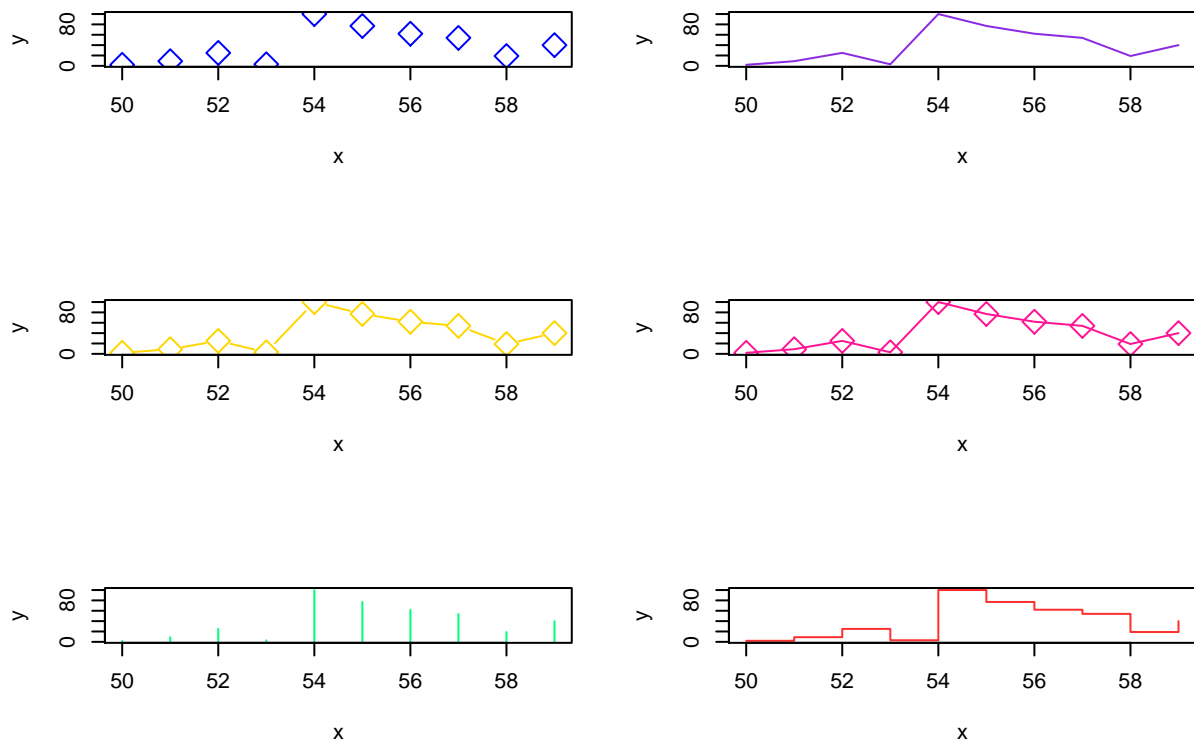


Figure 3: Segundo grupo de parámetros a considerar

```

par(mfrow = c(1,1))

```

Otros parámetros

- `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

Ejemplos

```
x = (2*(1:20))
y = (-1)^(1:20)*5*(1:20)
plot(x,y, main = "Ejemplo de gráfico", pch = 8,
     cex = 0.75, type = "b", lty = 4, lwd = 4,
     xaxp = c(0,40,2), yaxp = c(-100,100, 8))
```

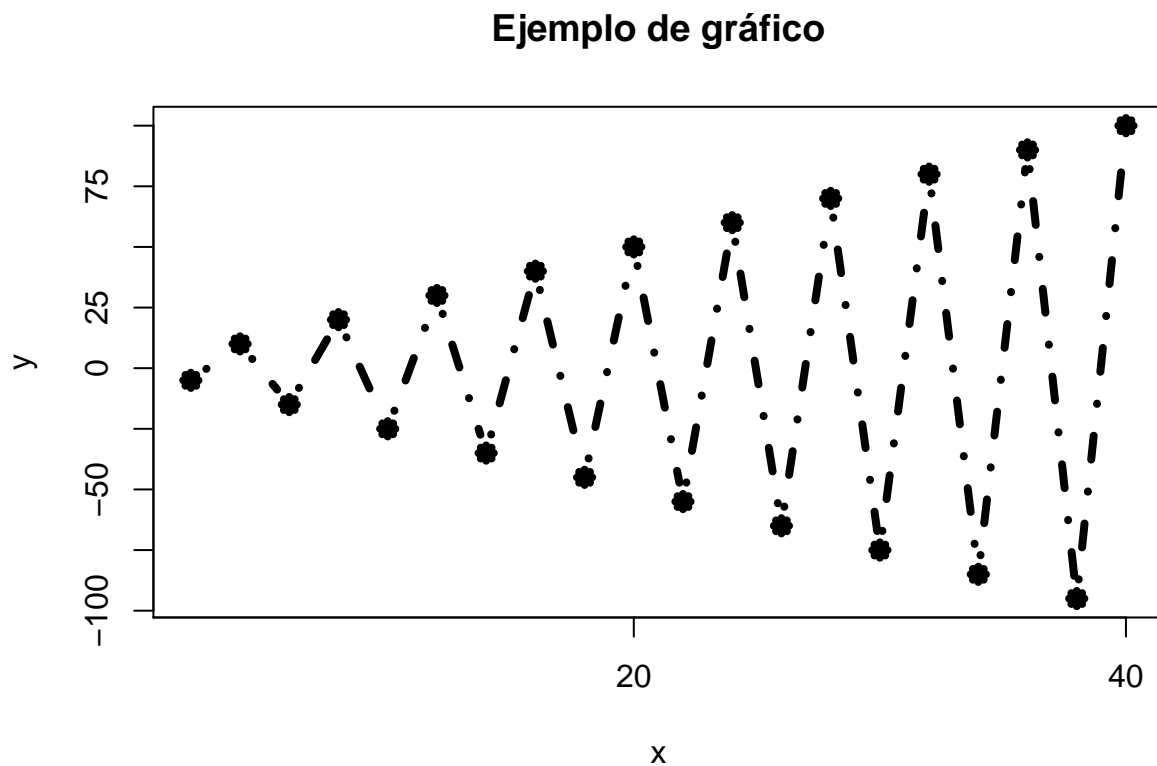


Figure 4: Tercer grupo de parámetros a considerar

```
plot(f, xlab = "n", ylab = expression(F[n]), main = "Secuencia de Fibonacci",
     pch = 6, cex = 0.75, col = "cadetblue4", lty = 2, lwd = 2,
     xlim = c(0,10), ylim = c(0,100), xaxp = c(0,10,2), yaxp = c(0,100,10),
     type = "b")
```

Elementos de un gráfico

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

Secuencia de Fibonacci

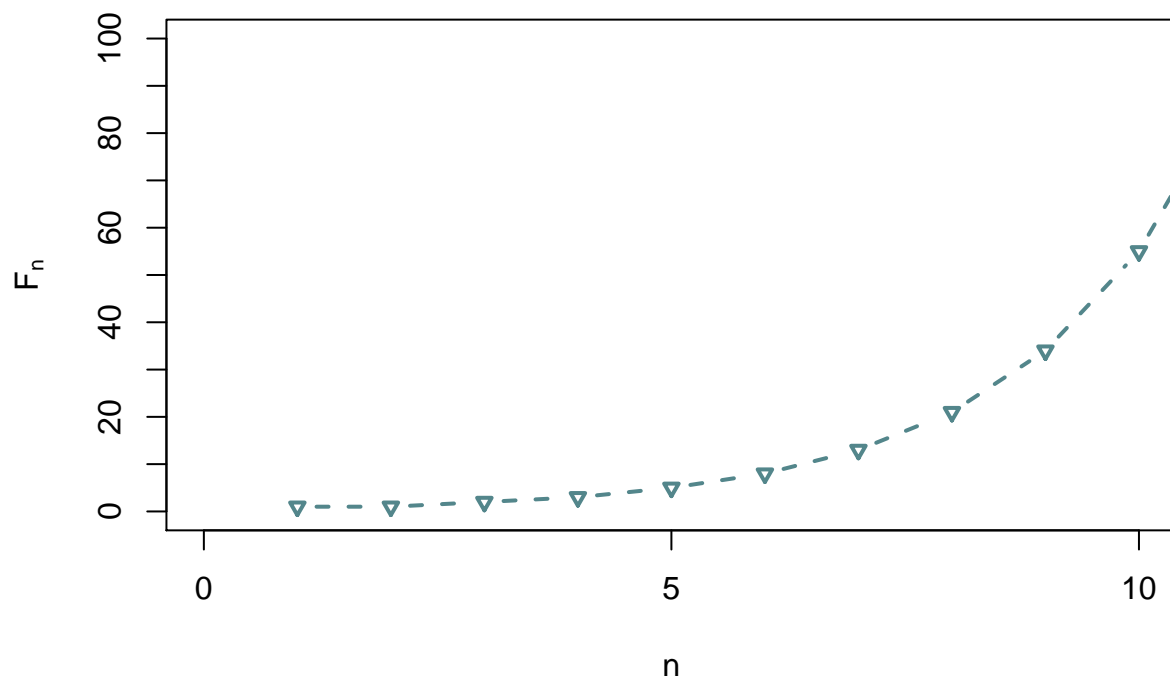


Figure 5: Secuencia de fibonacci limitando la gráfica

- `abline`: para añadir una recta a un gráfico ya existente
 - `abline(a,b)`: añade la recta $y = bx + a$
 - `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

Ejemplo

```
f = function(x){x^2}
plot(f, xlim = c(-3,3), col = "blue", xlab = "x", ylab = expression(x^2))
points(-3:3, (-3:3)^2, col = "forestgreen", pch = 5)
abline(v = -3:3, h = 0:9, col = "gray", lty = 2)
```

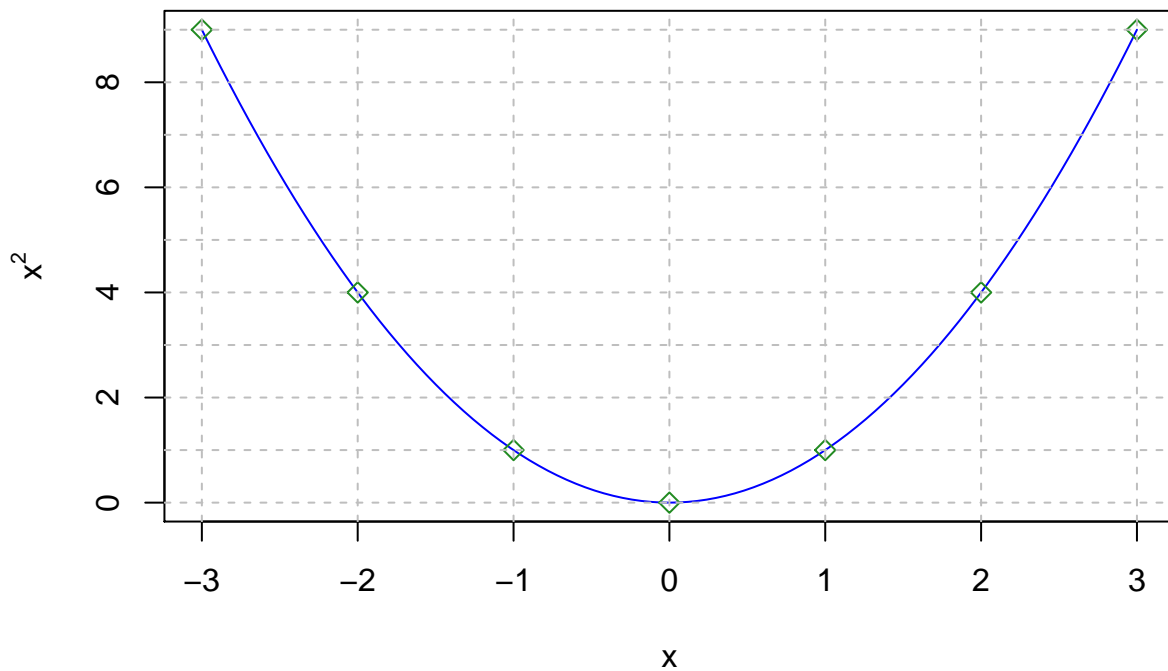


Figure 6: Uso de elementos de un gráfico

Para añadir texto a un punto en específico

- `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: por defecto: texto centrado

```
plot(alumnos, notas, ylim = c(0,10), main = "Notas de alumnos", col = "blue")
abline(v = 1:10, h = 0:10, col = "gray", lty = 2)
text(alumnos,
     notas,
     labels = c("A","B","C","D","E","F","G","H","I","J"),
     pos = c(rep(3,times = 4), 1, rep(3,times = 5)))
```

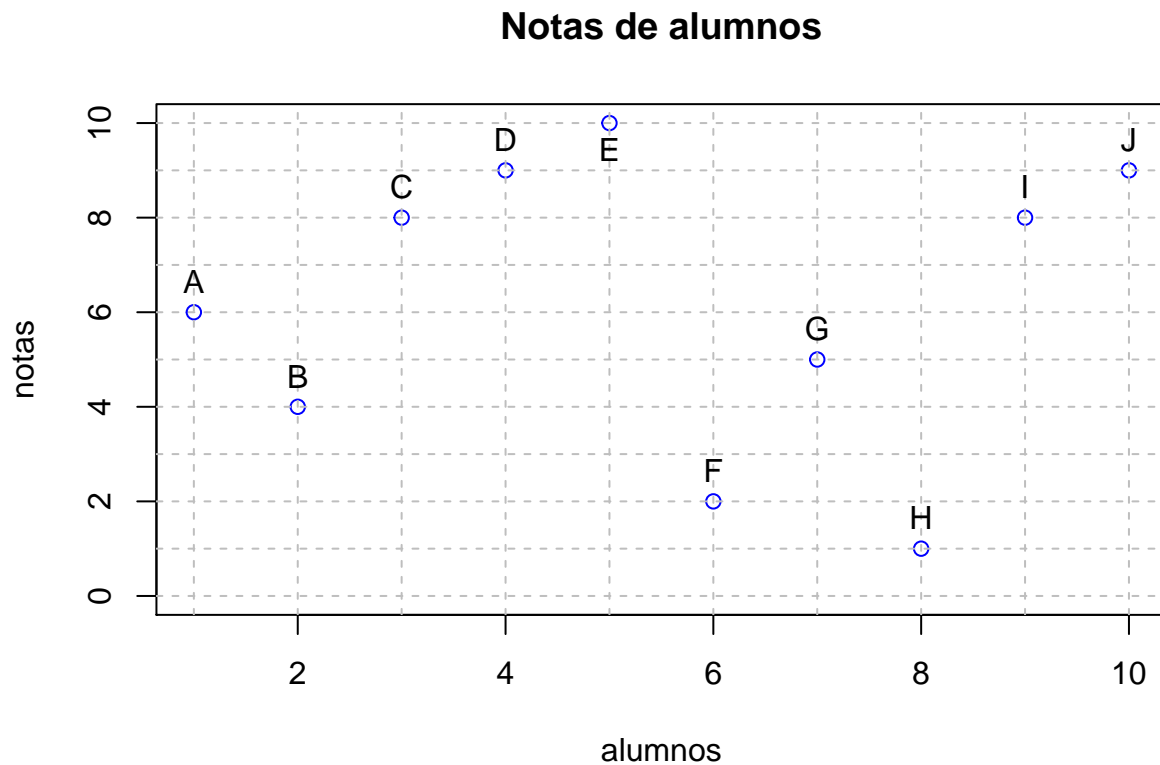
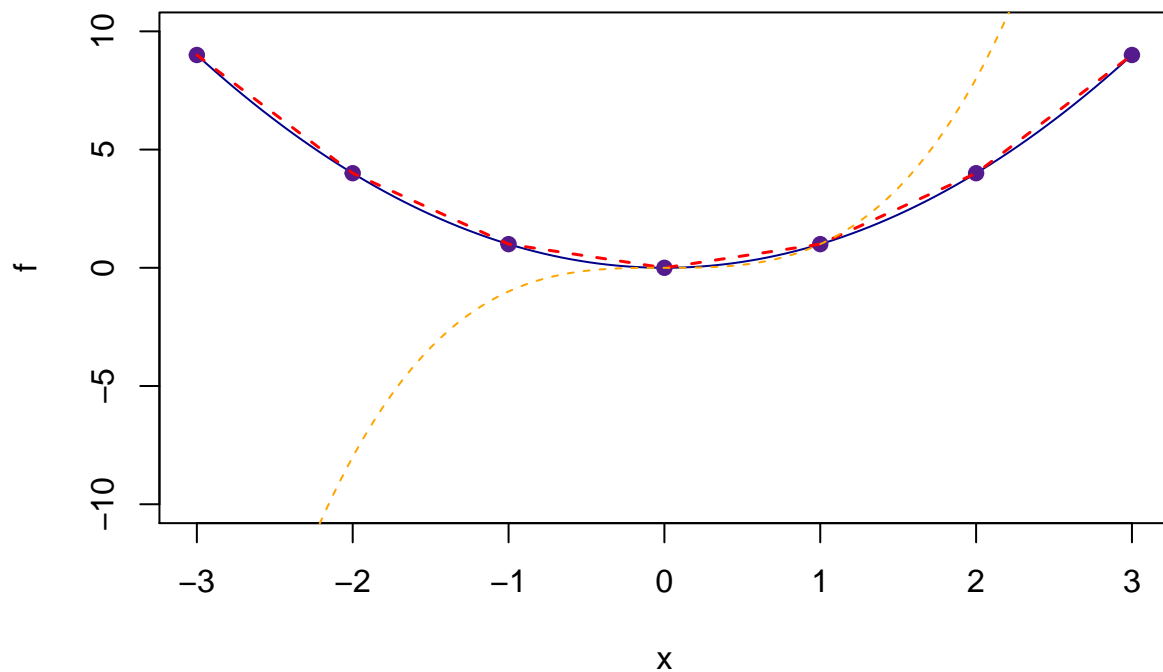


Figure 7: Ejemplo de uso de text()

Para añadir más líneas a nuestra gráfica

- `lines(x,y)`: añade a un gráfico existente una línea poligona lque une los puntos (x,y) 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

```
plot(f, col = "blue4", xlim = c(-3,3), ylim = c(-10,10))
points(-3:3, (-3:3)^2, pch = 19, col = "purple4")
lines(-3:3, (-3:3)^2, lty = "dashed", col = "red", lwd = 1.5)
curve(x^3, add = TRUE, lty = "dashed", col = "orange")
```



Añadiendo una leyenda

- `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

```
x = seq(-2*pi, 2*pi, 0.1)
plot(x, sin(x), col = "skyblue3", type = "l", lwd = 2)
lines(x, cos(x), col = "yellow3", lwd = 2)
lines(x, tan(x), col = "olivedrab3", lwd = 2)
legend("bottomleft", col = c("skyblue3", "yellow3", "olivedrab3"), legend = c("Seno", "Coseno", "Tangente"))
```

