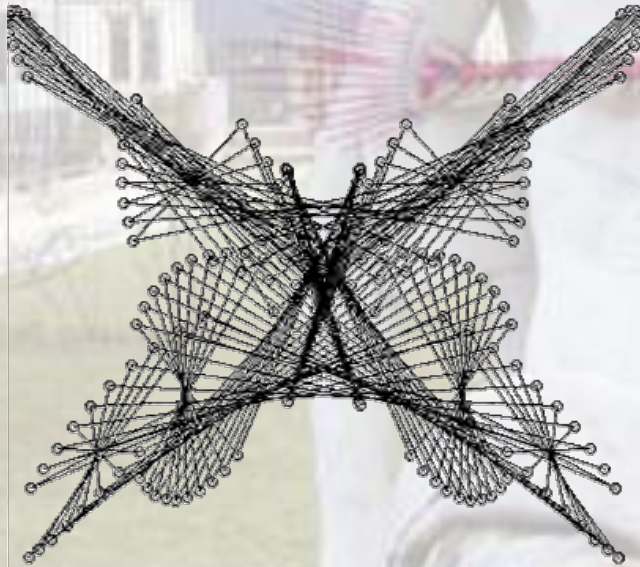


LABORATORIO DE ANATOMÍA ANIMAL

INGENIERIA INVERSA APLICADA A LA ANATOMÍA ANIMAL

M O O C



**22.- Análisis de datos y gráficas usando R:
Curvas planas.**



<http://cran.r-project.org/>

Software libre

A screenshot of the RGui (64-bit) application window. The title bar reads "RGui (64-bit)". Below the title bar is a menu bar with "Archivo", "Editar", "Visualizar", "Misc", "Paquetes", "Ventanas", and "Ayuda". Below the menu bar is a toolbar with icons for file operations (open, save, print, etc.). The main window is titled "R Console" and contains the following text:

```
R version 3.0.1 (2013-05-16) -- "Good Sport"  
Copyright (C) 2013 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R es un software libre y viene sin GARANTIA ALGUNA.  
Usted puede redistribuirlo bajo ciertas circunstancias.  
Escriba 'license()' o 'licence()' para detalles de distribucion.  
  
R es un proyecto colaborativo con muchos contribuyentes.  
Escriba 'contributors()' para obtener más información y  
'citation()' para saber cómo citar R o paquetes de R en publicaciones.  
  
Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,  
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.  
Escriba 'q()' para salir de R.  
  
> |
```

Uso de R como calculadora:

```
> 4+2
[1] 6
> 5*2 #multiplicación
[1] 10
> 4+2-5 # suma y resta
[1] 1
> pi #número pi
[1] 3.141593
> 2*pi*5 #circunferencia
[1] 31.41593
> n<-5 #asignando valores
> 2*pi*n
[1] 31.41593
> n<- -5 #aisgnando valores negativos
> n
[1] -5
> 3*4^2; (3*4)^2 # potencias
[1] 48
[1] 144
> |
```

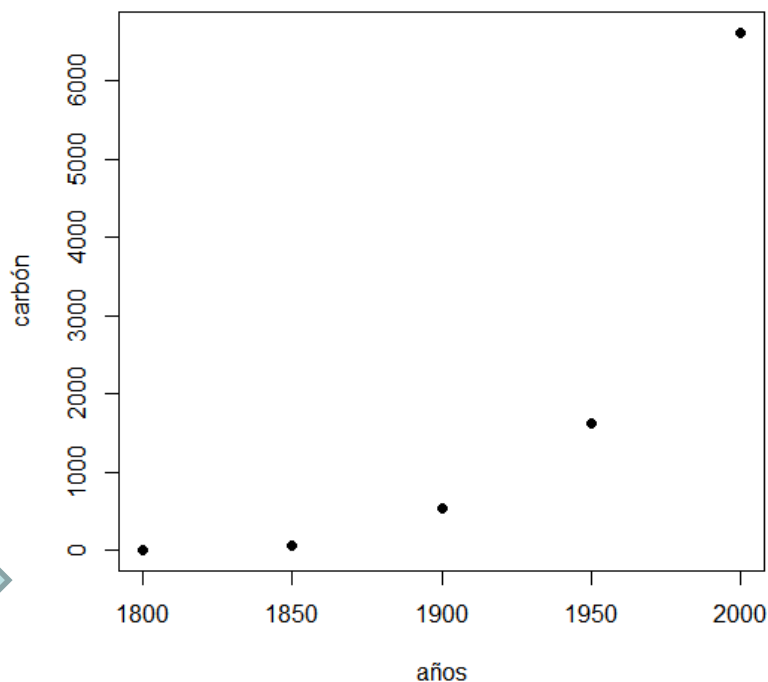
El [1] significa que el resultado es un número

```
> 4+2
[1] 6
> 5*2 #multiplicación
[1] 10
> 4+2-5 # suma y resta
[1] 1
> pi #número pi
[1] 3.141593
> 2*pi*5 #circunferencia
[1] 31.41593
> n<-5 #asignando valores
> 2*pi*n
[1] 31.41593
> n<- -5 #aisgnando valores negativos
> n
[1] -5
> 3*4^2; (3*4)^2 # potencias
[1] 48
[1] 144
```

```
> #entrada de datos en linea
> años <- c(1800, 1850, 1900,1950, 2000)
> carbón <- c(8,54, 534,1630,6611)
> plot (carbón ~ años, pch=16)
> |
```

```
> gasolina <-data.frame(años,carbón)
> gasolina
  años carbón
1 1800      8
2 1850     54
3 1900    534
4 1950   1630
5 2000   6611
```

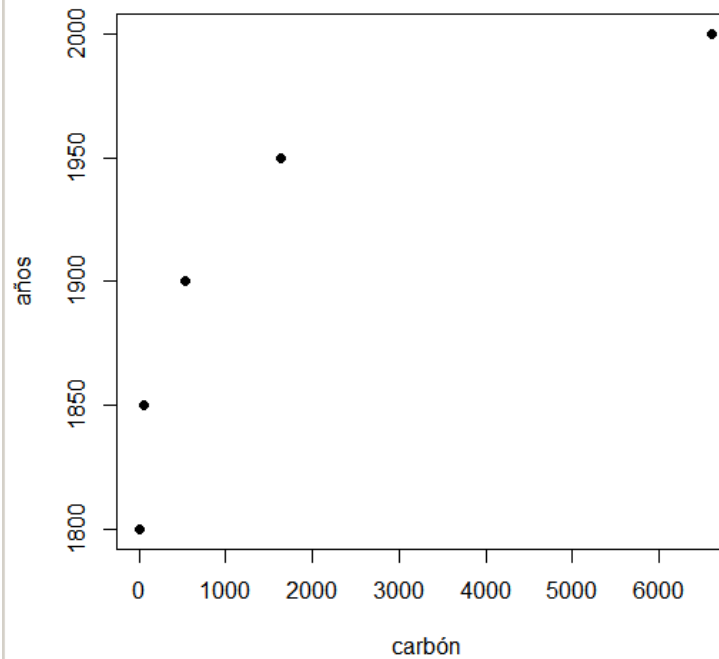
```
> plot (gasolina, pch=16)
> |
```



```

> gasolina <-data.frame(carbón,años)
> gasolina
  carbón años
1      8 1800
2     54 1850
3    534 1900
4   1630 1950
5   6611 2000
> plot (gasolina, pch=16)
> range (gasolina$carbón)
[1] 8 6611
> range (gasolina$años)
[1] 1800 2000
> sum(gasolina$años)
[1] 9500
> sum (gasolina$carbón)
[1] 8837
> mean (gasolina$carbón)
[1] 1767.4
> |

```

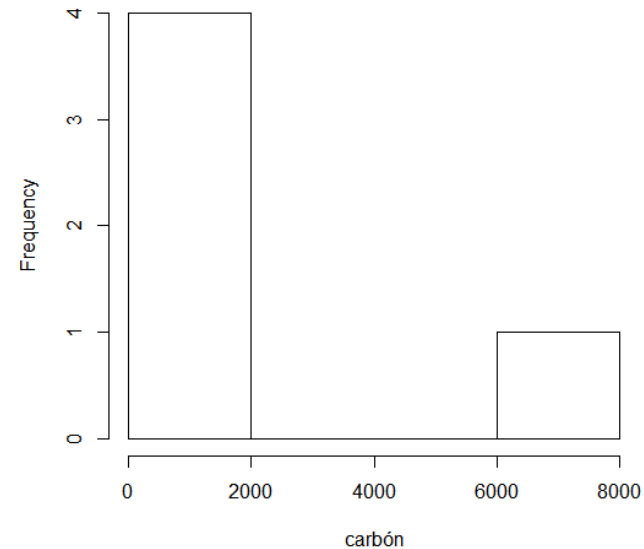


```

> sort(gasolina$carbón)
[1] 8 54 534 1630 6611
> length (gasolina$carbón)
[1] 5
> summary (carbón)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     8     54     534   1767   1630   6611
> |

```

Histogram of carbón



```

> hist (carbón)
> |

```



```

> #R es un lenguaje de programación interactivo
> celsius <- (0:4)*10 #matriz de 5 columnas
> celsius
[1] 0 10 20 30 40
> fahrenheit <- 9/5*celsius+32 # formula de conversión
> conversion <- data.frame(Tcelsius=celsius, Tfahrenheit=fahrenheit)
> print (conversion)
  Tcelsius Tfahrenheit
1         0          32
2        10          50
3        20          68
4        30          86
5        40         104
> |
> help (plot)
starting httpd help server ... done

```

```

> #VECTORES
> c(5,6,7,8,9,-3,10,-4)
[1] 5 6 7 8 9 -3 10 -4
> x <-c(5,6,7,8,9,-3,10,-4)
> y <-c(20,30,40)
> z <-c(x,y)
> z
[1] 5 6 7 8 9 -3 10 -4 20 30 40
> |

```

```

> #extracción de elementos
> z[c(2,4)] #extracción de los elementos que ocupan las posiciones 2 y 4
[1] 6 8
> z [-c (2,4)] # borra esos elementos
[1] 5 7 9 -3 10 -4 20 30 40
> |

```

Pero ojo

```

> z
[1] 5 6 7 8 9 -3 10 -4 20 30 40
> |
> v <- z[z>10]
> v
[1] 20 30 40
> |

```

parcela {gráficos}

RI

Generic XY Trazado

Descripción

Función genérica para el trazado de R objetos. Para más detalles sobre los argumentos gráficos de los parámetros, consulte [el par.](#)

Para gráficos de dispersión simples, [plot.default](#) se utilizarán. Sin embargo, existen [parcelas](#) métodos para muchos R objetos, incluyendo [la función de](#) s, [la densidad de](#) objetos, etc Use métodos (de la trama) y la documentación de los mismos.

Uso

```

> #comparando elementos de los vectores
> z > 10 #busca verdadero T o falso F
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE
> z [z>10]
[1] 20 30 40
> |

```

```

> m=matrix (1:12,3,4)
> m
  [,1] [,2] [,3] [,4]
[1,]  1   4   7  10
[2,]  2   5   8  11
[3,]  3   6   9  12
> m[2,4]
[1] 11
> m[2,]
[1] 2 5 8 11
> m[,4]
[1] 10 11 12
> |

```

```

> #Comparación de elementos de un vector
> x <-c(1,2,3,4,5,6,7,8,9)
> x
[1] 1 2 3 4 5 6 7 8 9
> x>5
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE
> x<= 5
Error: inesperado '<' in "x<="
> x<= 5
[1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE
> x>= 5
[1] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
> x != 5
[1] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE
> x == 5
[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
> |

```

```

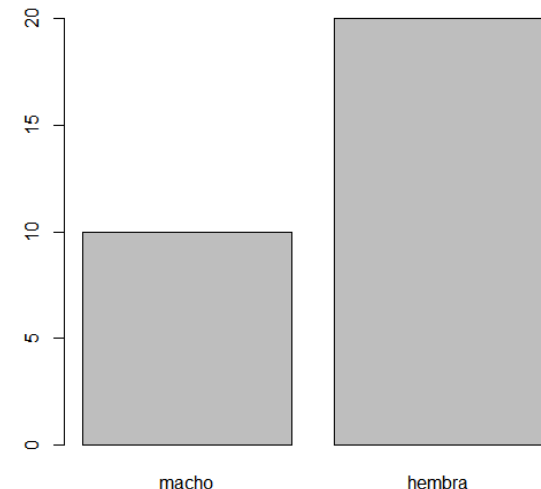
> #Modelado de datos
> 5:15
[1] 5 6 7 8 9 10 11 12 13 14 15
> 15:5
[1] 15 14 13 12 11 10 9 8 7 6 5
> seq (from=3,to=20, by=4)
[1] 3 7 11 15 19
> seq (3,20,4)
[1] 3 7 11 15 19
> rep (c(3,20,4),4) #repite la secuencia
[1] 3 20 4 3 20 4 3 20 4 3 20 4
> |

```

```

> #factores
> genero <-c(rep("hembra", 20),rep("macho",10))
> genero
[1] "hembra" "hembra" "hembra" "hembra" "hembra" "hembra" "hembra" "hembra"
[9] "hembra" "hembra" "hembra" "hembra" "hembra" "hembra" "hembra" "hembra"
[17] "hembra" "hembra" "hembra" "hembra" "macho" "macho" "macho" "macho"
[25] "macho" "macho" "macho" "macho" "macho" "macho"
> levels (genero)
NULL
> genero <-factor(genero)
> levels(genero)
[1] "hembra" "macho"
> |
> genero <- factor(genero,levels=c("macho","hembra"))
> genero
[1] hembra hembra hembra hembra hembra hembra hembra hembra hembra hembra hembra
[11] hembra hembra hembra hembra hembra hembra hembra hembra hembra hembra hembra
[21] macho macho macho macho macho macho macho macho macho macho macho
Levels: macho hembra
> plot (genero)
> |

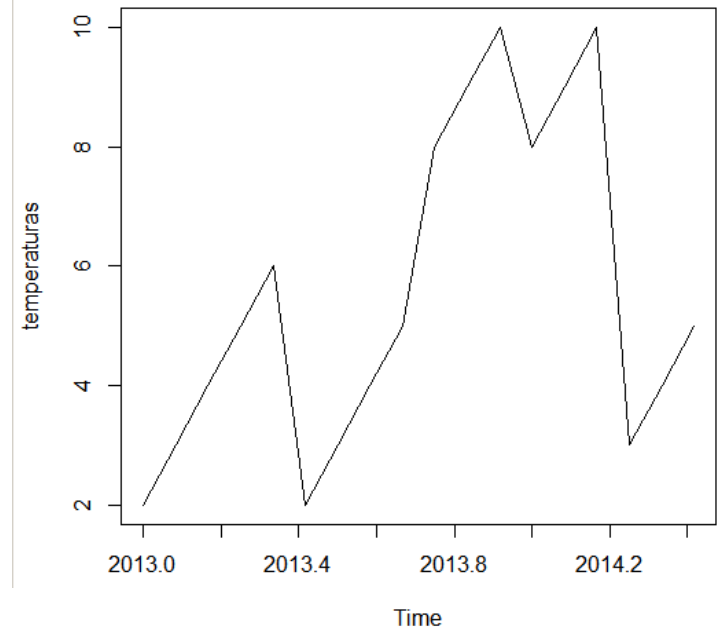
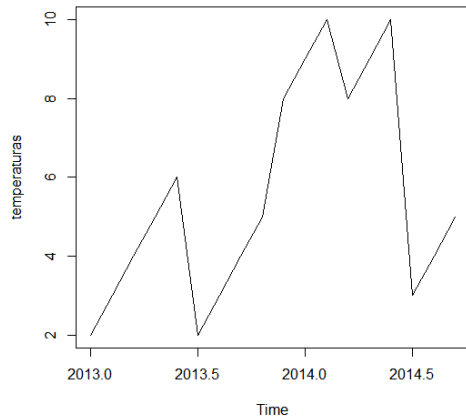
```



```

> #series temporales
> temperaturas <-c(2,3,4,5,6,2, 3 ,4 5, 8,9,10, 8,9,10,3,4,5)
Error: inesperado constante numérica in "temperaturas <-c(2,3,4,5,6,2, 3 ,4 5"
> temperaturas <-c(2,3,4,5,6,2, 3 ,4 5, 8,9,10, 8,9,10,3,4,5)
Error: inesperado constante numérica in "temperaturas <-c(2,3,4,5,6,2, 3 ,4 5"
> temperaturas <-c(2,3,4,5,6,2, 3 ,4,5,8,9,10,8,9,10,3,4,5)
> temperaturas
[1] 2 3 4 5 6 2 3 4 5 8 9 10 8 9 10 3 4 5
> temperaturas<-ts(temperaturas, start=2013, frequency=10)
> plot(temperaturas)
> temperaturas<-ts(temperaturas, start=2013, frequency=12)
> plot(temperaturas)
> |

```



```

> #funciones
> cat(temperaturas)
2 3 4 5 6 2 3 4 5 8 9 10 8 9 10 3 4 5>
> cumprod(temperaturas)
[1] 2.000000e+00 6.000000e+00 2.400000e+01 1.200000e+02 7.200000e+02
[6] 1.440000e+03 4.320000e+03 1.728000e+04 8.640000e+04 6.912000e+05
[11] 6.220800e+06 6.220800e+07 4.976640e+08 4.478976e+09 4.478976e+10
[16] 1.343693e+11 5.374771e+11 2.687386e+12
> cumsum(temperaturas)
[1] 2 5 9 14 20 22 25 29 34 42 51 61 69 78 88 91 95 100
> diff(temperaturas)
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2013      1  1  1  1  -4  1  1  1  3  1  1
2014  -2  1  1  -7  1  1
> length(temperaturas)
[1] 18

```

```

> mean(temperaturas)
[1] 5.555556
> median(temperaturas)
[1] 5
> range(temperaturas)
[1] 2 10

```



```

> #Función function(){}
> media.y.desviación <-function(x){ # nombre de la función
+ a<-mean(x) # cuerpo de la función
+ b<-sd(x)
+ c(Media=a,DS=b) #valor de retorno
+ }
> media.y.desviación (temperaturas)
      Media      DS
5.555556 2.748737
> |

```

```

> #Operador %in%
> temperaturas[temperaturas %in% c(2,5)]
[1] 2 5 2 5 5
> #Operador de coincidencia
> match (temperaturas,c(2,5), nomatch=0)
[1] 1 0 0 2 0 1 0 0 2 0 0 0 0 0 0 0 2

```

Guardar área de trabajo

```

> save.image("D:\\TRABAJO\\MOOC\\Materiales_07\\Ejemplos.RData")

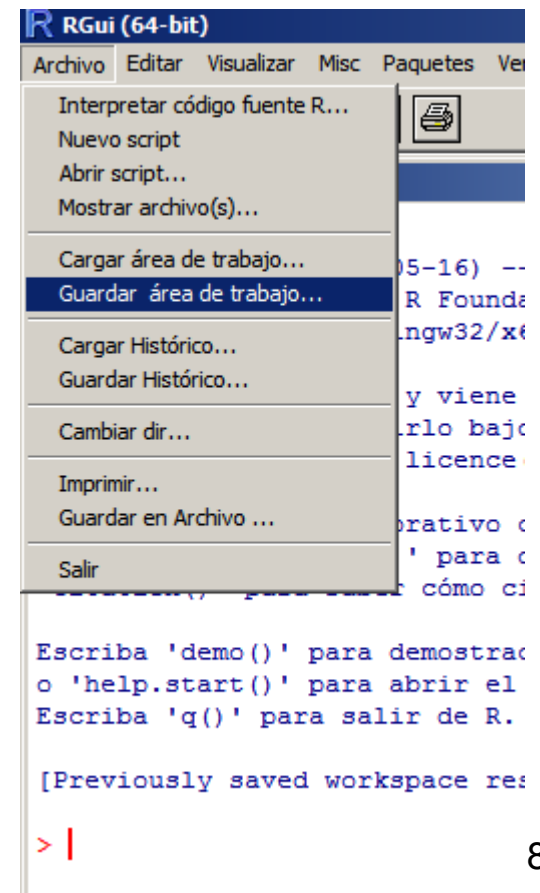
```

Recuperar área de trabajo

```

> load("D:\\TRABAJO\\MOOC\\Materiales_07\\Ejemplos.RData")
> temperaturas
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2013   2  3  4  5  6  2  3  4  5  8  9 10
2014   8  9 10  3  4  5
> |

```



Importar datos

Se copian de una table Excel por ejemplo

	K	L	M
	mm	gr	
	70	28.5	
	70	26.5	
	50	9.7	
	60	17	
	55	16	
	54	14	
	52	15	
	72	41	
	64	23	
	67	27	
	47	10	
	61	28	
	58	22	
	48	13	
	41	7	

```
> ranas <-read.table ("clipboard", header=T)
> ranas
  mm  gr
1  70 28.5
2  70 26.5
3  50  9.7
4  60 17.0
5  55 16.0
6  54 14.0
7  52 15.0
8  72 41.0
9  64 23.0
10 67 27.0
11 47 10.0
12 61 28.0
13 58 22.0
14 48 13.0
15 41  7.0
> |
```

```
> ranas <-read.table ("clipboard")
```

```
Mensajes de
In read.tab
incomplet
> ranas <-r
> ranas
  mm  gr
1  70 28.5
2  70 26.5
3  50  9.7
4  60 17.0
5  55 16.0
6  54 14.0
7  52 15.0
8  72 41.0
9  64 23.0
10 67 27.0
11 47 10.0
12 61 28.0
13 58 22.0
14 48 13.0
15 41  7.0
> fix(ranas
```

R Editor de datos				
	mm	gr	var3	var4
1	70	28.5		
2	70	26.5		
3	50	9.7		
4	60	17		
5	55	16		
6	54	14		
7	52	15		
8	72	41		
9	64	23		
10	67	27		
11	47	10		
12	61	28		
13	58	22		
14	48	13		
15	41	7		
16				

Editar Visualizar Misc Paquetes Ventanas Ayuda

- Copiar Ctrl+C
- Pegar Ctrl+V
- Pegar solo comandos
- Copiar y Pegar... Ctrl+X
- Seleccionar todo
- Limpar consola
- Editor de datos...
- Preferencias de la interface gráfico

R Console

```
> 417
[1] 417
> ranas <-read.t
Mensajes de avis
In read.table("c
incomplete fin
> ranas <-read.t
> ranas
  mm  gr
1  70 28.5
2  70 26.5
```

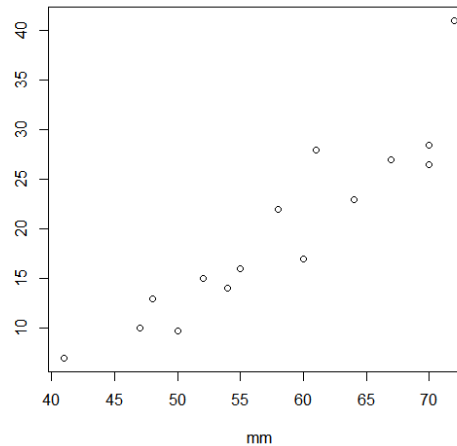
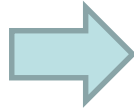
Question

Nombre de data fram

ranas

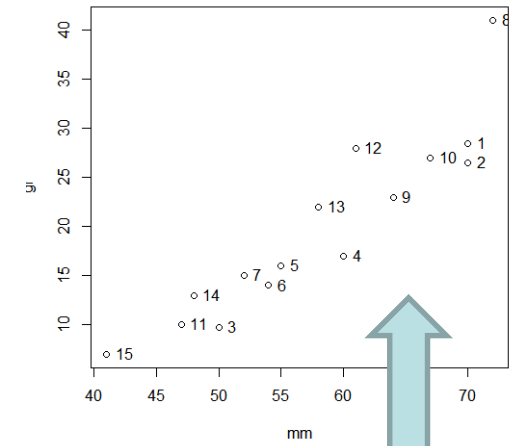
Gráficos en R

```
> with(ranas, plot(mm, gr))
```



```
> with(ranas, text(mm, gr, labels=row.names(ranas), pos=4))
```

Pos= 4 situa el texto a la derecha de los puntos



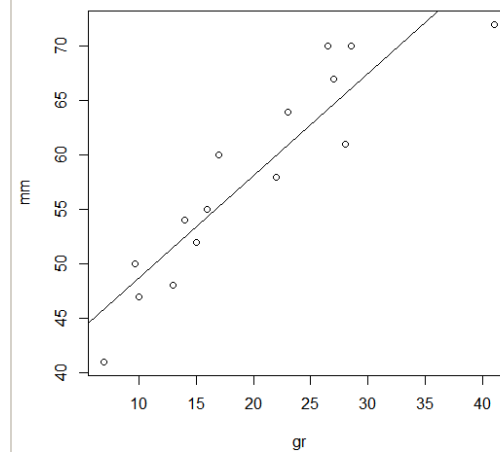
Análisis de datos

```
> with(ranas, cor(mm,gr)) #correlaciones
[1] 0.9200672
> with(log(ranas),cor(mm,gr))#correlación logarítmica
[1] 0.9517723
> with (ranas, cor (mm,gr,method="spearman"))
[1] 0.9526367
> |
```

```
> Linea.lm<-lm(mm~gr, data=ranas)
> plot (mm~gr, data=ranas)
> abline(Linea.lm)
> |
```

```
> coef(Linea.lm)
(Intercept)      gr
39.2360890    0.9420849
> round (fitted (Linea.lm),1)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
66.1 64.2 48.4 55.3 54.3 52.4 53.4 77.9 60.9 64.7 48.7 65.6 60.0 51.5 45.8
> round (resid (Linea.lm),1)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 3.9  5.8  1.6  4.7  0.7  1.6 -1.4 -5.9  3.1  2.3 -1.7 -4.6 -2.0 -3.5 -4.8
> Linea.lm$coef
(Intercept)      gr
39.2360890    0.9420849
```

```
> with (ranas, cor (mm,gr,method="kendall"))
[1] 0.8421149
```



```
> ranas <-log(ranas)
> ranas
      mm      gr
1 4.248495 3.349904
2 4.248495 3.277145
3 3.912023 2.272126
4 4.094345 2.833213
5 4.007333 2.772589
6 3.988984 2.639057
7 3.951244 2.708050
8 4.276666 3.713572
9 4.158883 3.135494
10 4.204693 3.295837
11 3.850148 2.302585
12 4.110874 3.332205
13 4.060443 3.091042
14 3.871201 2.564949
15 3.713572 1.945910
> plot (mm~gr, data=ranas)
> |
```

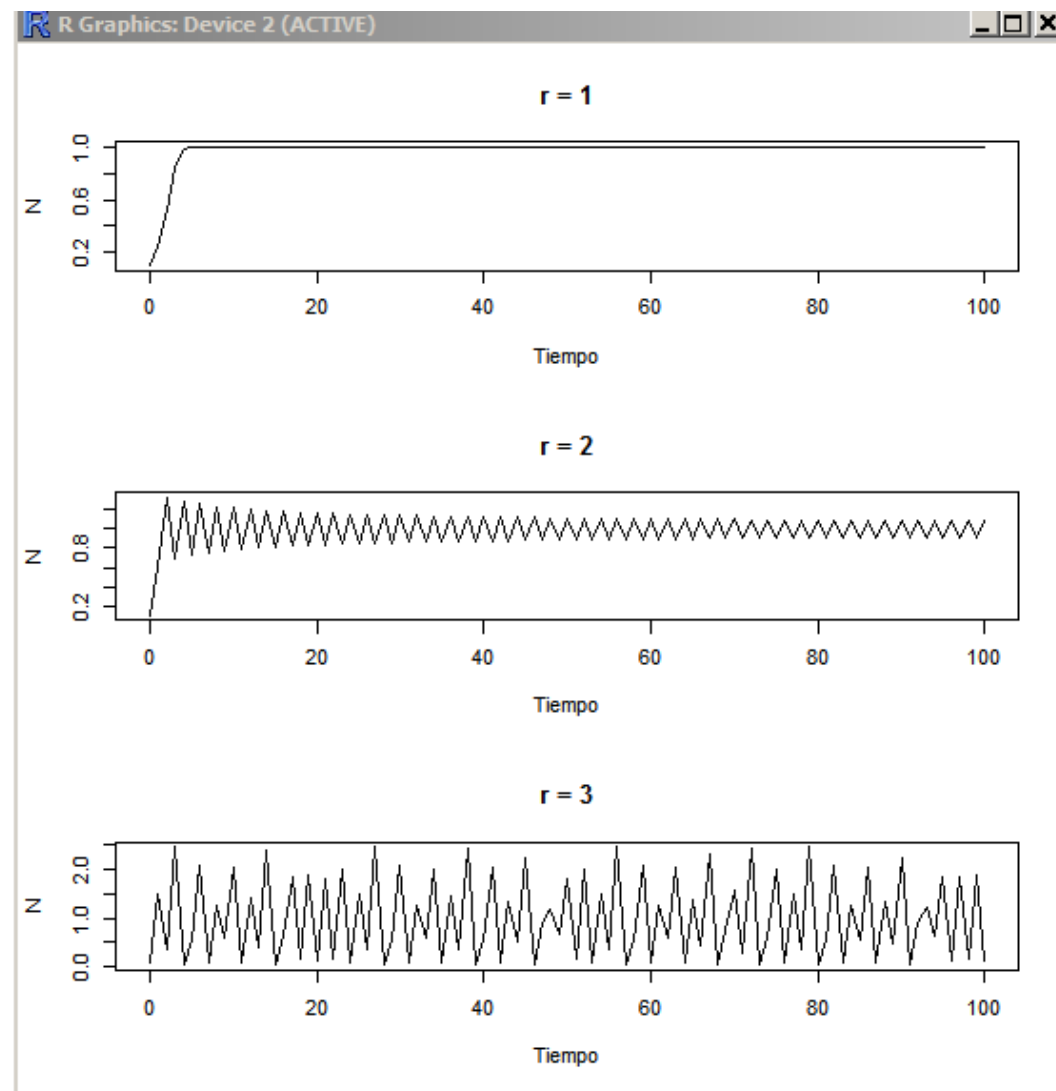
```

> ricker <- function(nzero, r, K=1, tiempo=100, desde=0, hasta=tiempo)
+ {
+ N <- numeric(tiempo+1)
+ N[1] <- nzero
+ for (i in 1:tiempo) N[i+1] <- N[i]*exp(r*(1 - N[i]/K))
+ Tiempo <- 0:tiempo
+ plot(Tiempo, N, type="l", xlim=c(desde, hasta))
+ }
> layout(matrix(1:3, 3, 1))
> ricker(0.1, 1); title("r = 1")
> ricker(0.1, 2); title("r = 2")
> ricker(0.1, 3); title("r = 3")
> |

```

Curvas planas

Ver
MOOC_022_complementos:
Ejemplos_022.txt



Se llama dominio de existencia “D” al conjunto de valores de x para los que la $f(x)$ tiene un valor bien determinado.

Utilizaremos las expresiones:

$D = \mathbb{R}$ se define como para todo x perteneciente a \mathbb{R}

$D = \mathbb{R} - \{0\}$ para todo x perteneciente a \mathbb{R} excepto el 0

$D = [0, +\infty[$ para todo x entre 0 e infinito

$D =]0, +\infty[$ para todo x mayor de 0 hasta infinito

$D =]-2, +2[$ para todo x mayor de -2 y menor de +2

FUNCIONES ALGEBRAICAS

Función racional entera o polinómica

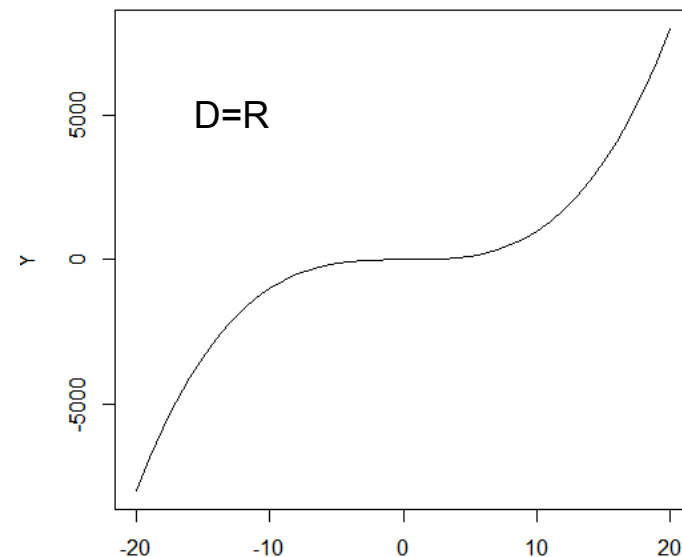
$$y=x^3$$

```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-desde^3
+ X[i] <-desde
+ desde=desde+1
+ }
>
> plot (X,Y,type="l")
> title (main="función potencial")
> |
```

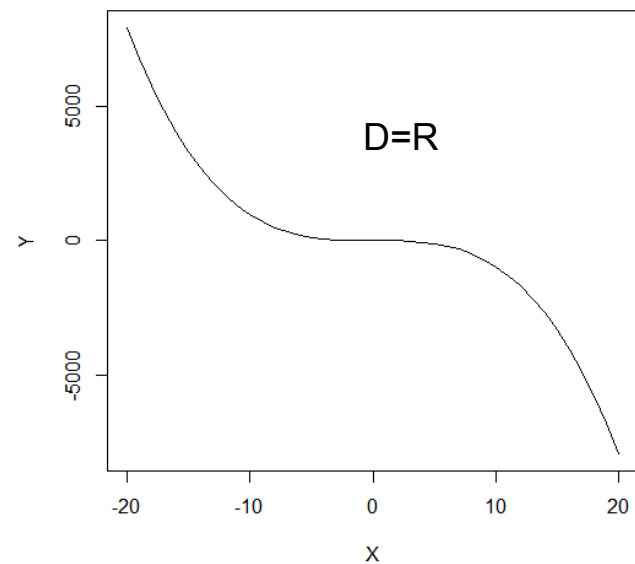
```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-3*desde-(desde^3)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función potencial")
```

$$y = 3x - x^3$$

función potencial



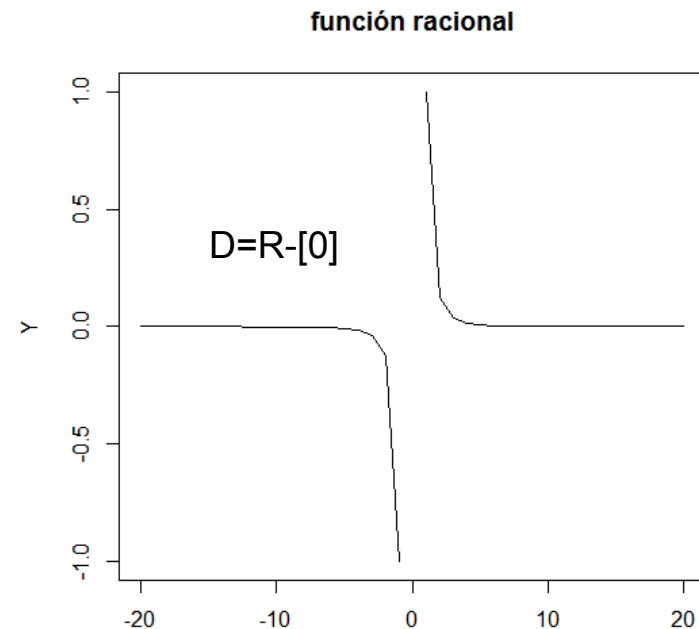
función potencial



Función racional

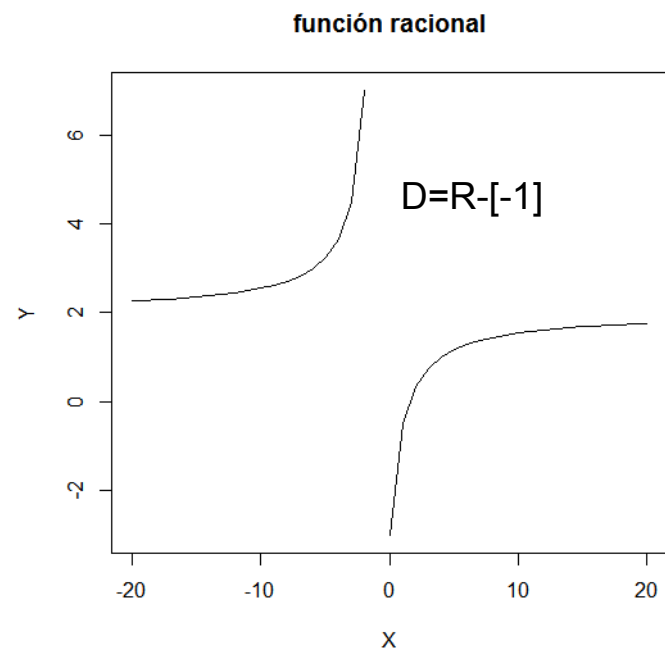
```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-1/(desde^3)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función racional")
```

$$y = \frac{1}{x^3}$$



```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <- (2*desde-3)/(desde+1)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función racional")
> |
```

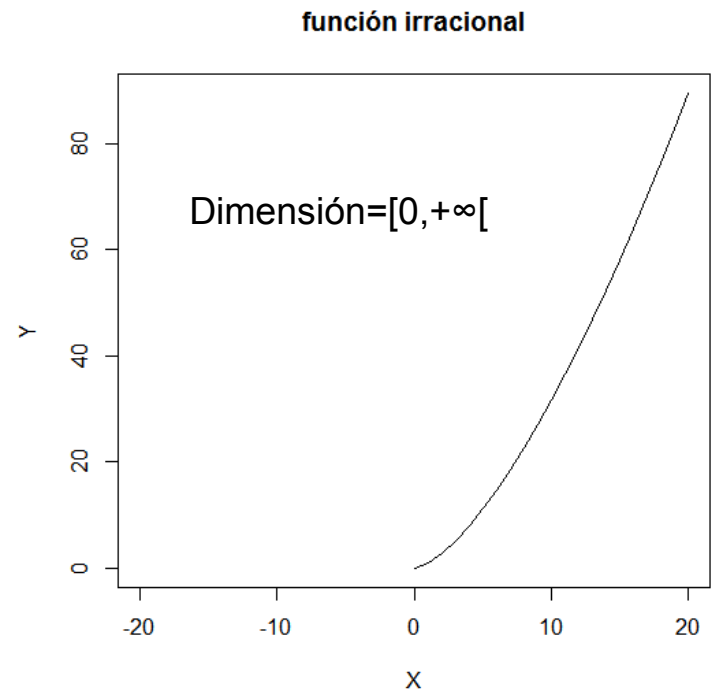
$$y = \frac{2X - 3}{X + 1}$$



Función irracional

$$y = x^{3/2}$$

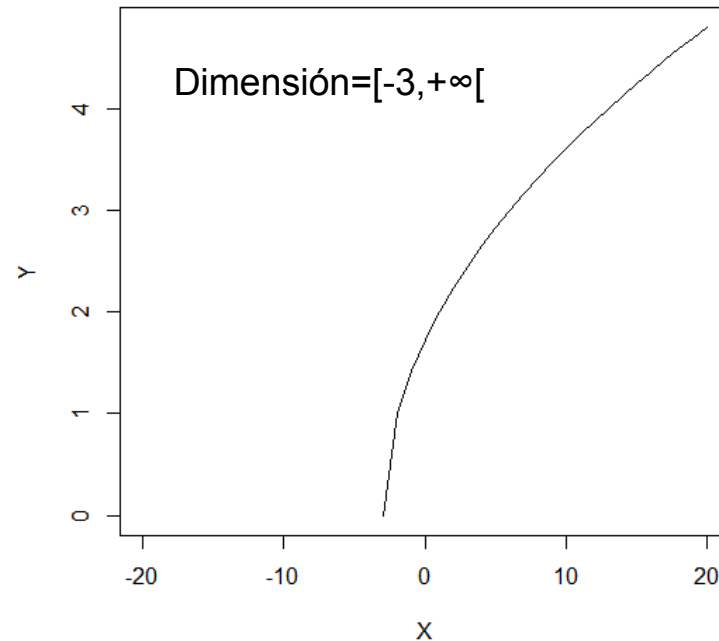
```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-desde^(3/2)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función irracional")
|
```



```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-sqrt(desde+3)
+ X[i] <-desde
+ desde=desde+1
+ }
Hubo 17 avisos (use warnings() para verlos)
> warnings()
Warning messages:
1: In sqrt(desde + 3) : Se han producido NaNs
2: In sqrt(desde + 3) : Se han producido NaNs
3: In sqrt(desde + 3) : Se han producido NaNs
4: In sqrt(desde + 3) : Se han producido NaNs
5: In sqrt(desde + 3) : Se han producido NaNs
6: In sqrt(desde + 3) : Se han producido NaNs
7: In sqrt(desde + 3) : Se han producido NaNs
8: In sqrt(desde + 3) : Se han producido NaNs
9: In sqrt(desde + 3) : Se han producido NaNs
```

$$y = \sqrt{x + 3}$$

función irracional



FUNCIONES TRASCENDENTES

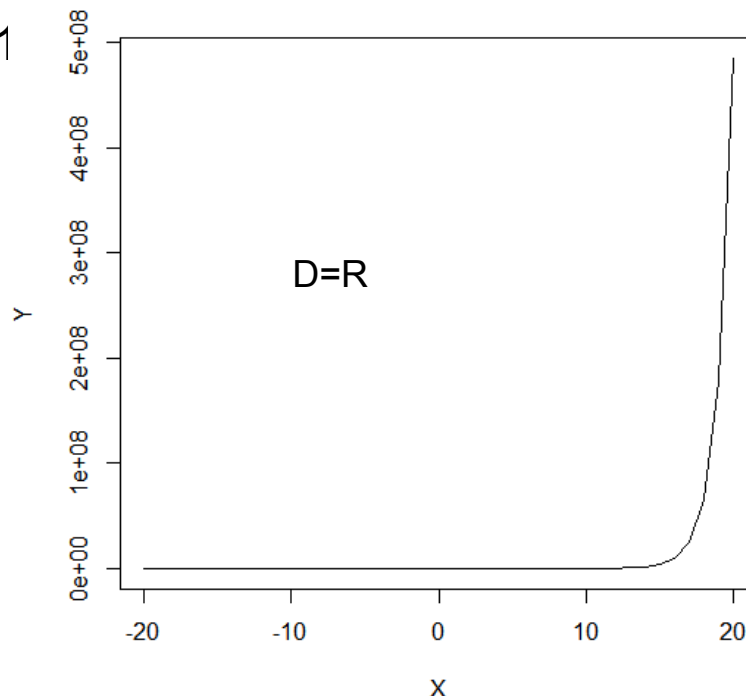
Exponencial

$$y = a^x, a > 0, a \neq 1$$

```
> # Trascendente exponencial
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-exp(1)^desde
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función exponencial")
> #####
> exp(1)
[1] 2.718282
```

a=e

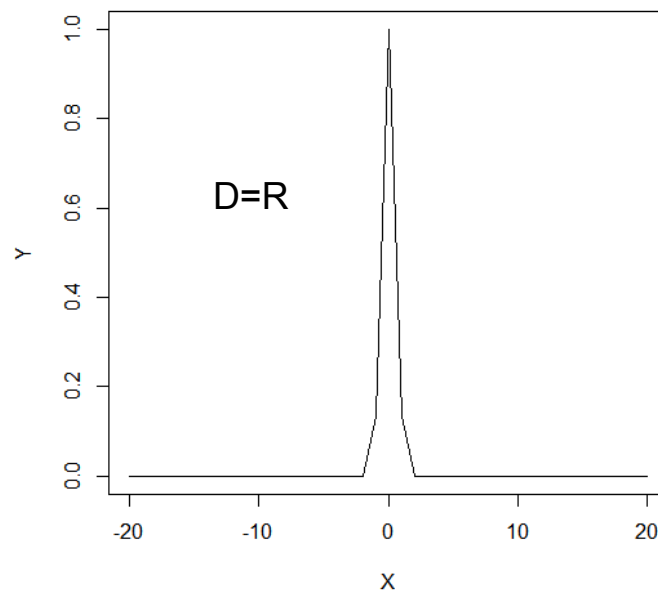
función exponencial



$$y = e^{-2x^2}$$

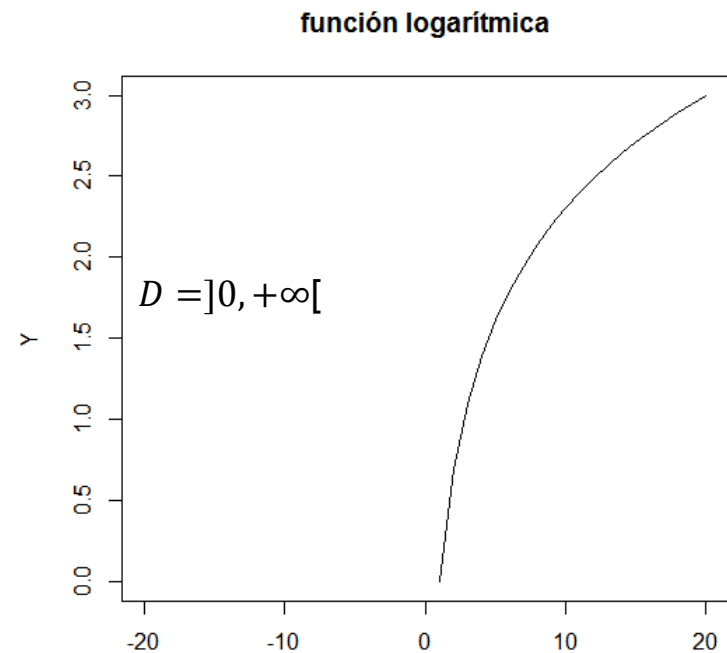
```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-exp(1)^(-2*(desde^2))
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función exponencial")
> |
```

función exponencial



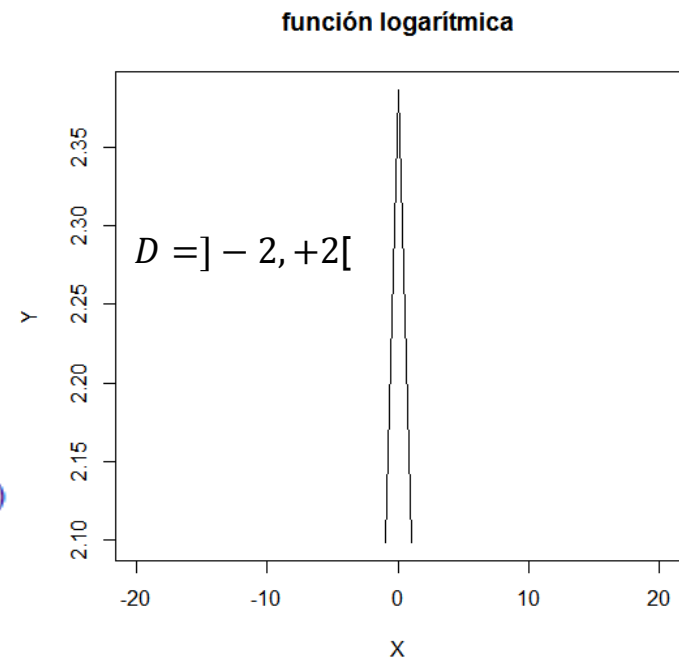
Logarítmica $y = \ln x$

```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-log(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
Hubo 20 avisos (use warnings() para verlos)
> plot (X,Y,type="l")
> title (main="función logarítmica")|
```



$$y = \ln(4 - x^2) + 1$$

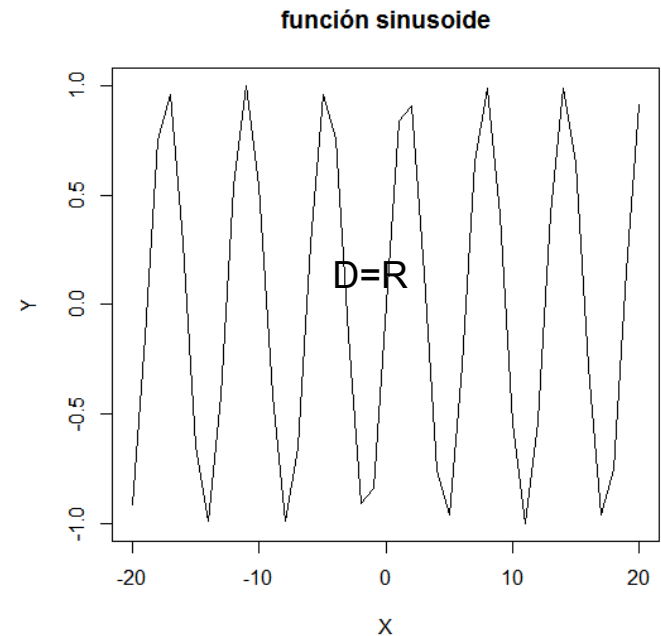
```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-log(4-desde^2)+1
+ X[i] <-desde
+ desde=desde+1
+ }
Hubo 36 avisos (use warnings() para verlos)
> plot (X,Y,type="l")
> title (main="función logarítmica")|
```



FUNCIONES TRIGONOMÉTRICAS

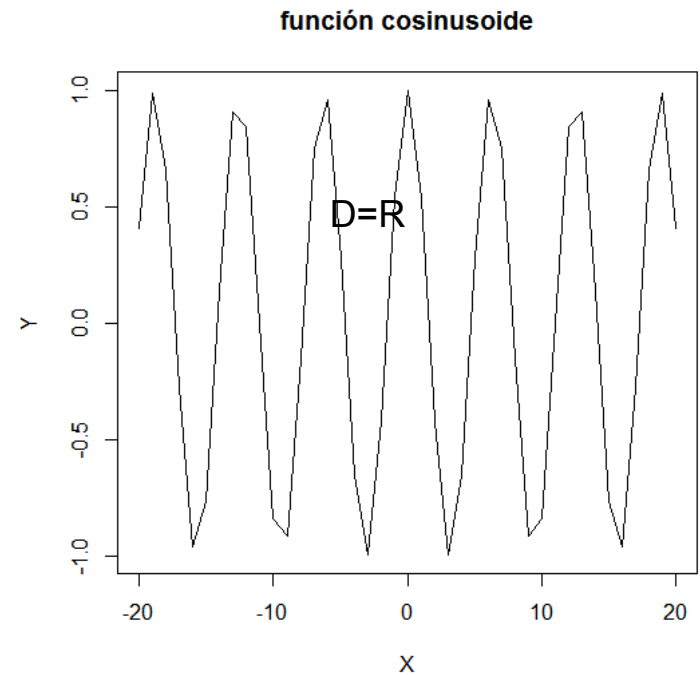
$$y = \text{sen}x$$

```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-sin(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función senoide")
. |
```



$$y = \text{cos}x$$

```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-cos(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función cosenoide")
. |
```

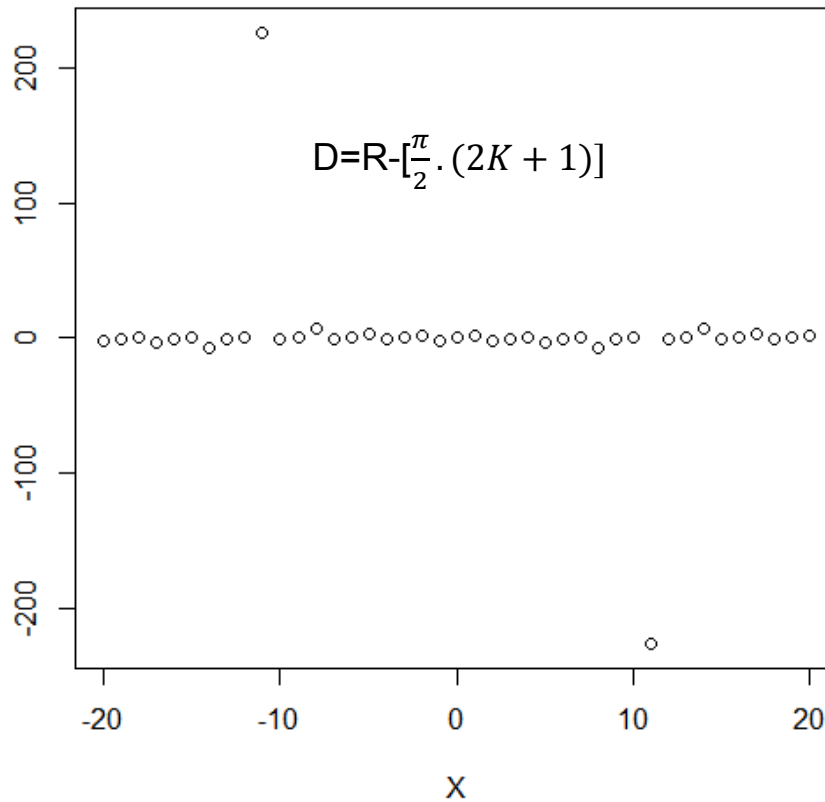
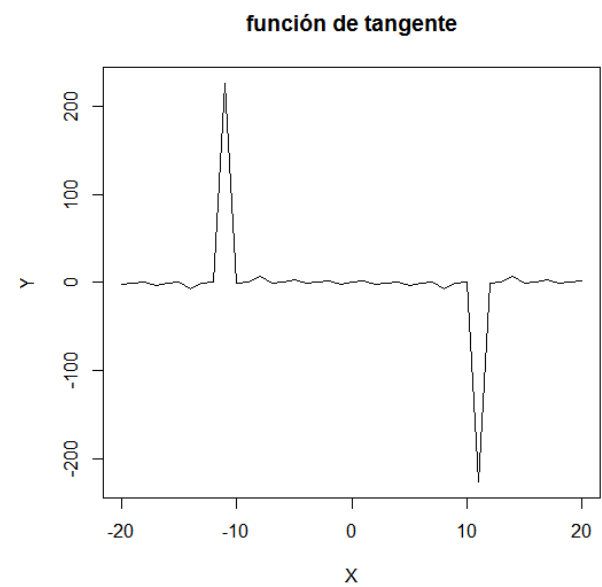


```

> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-tan(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y,type="l")
> title (main="función de tangente")

```

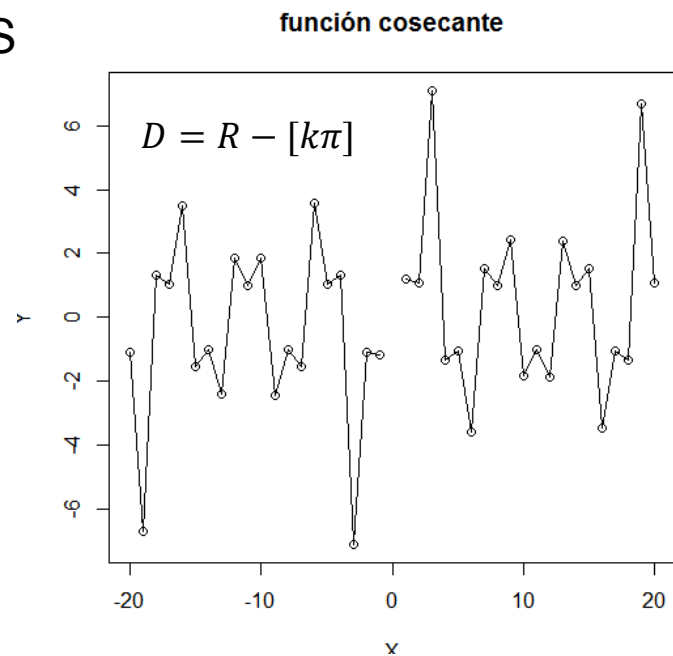
$$y = \operatorname{tg}x$$



FUNCIONES TRIGONOMÉTRICAS INVERSAS

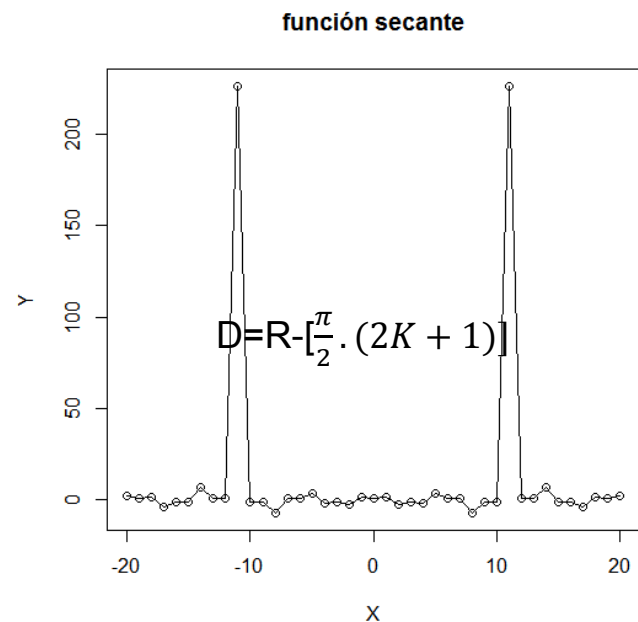
```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-1/sin(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función cosecante")
-----
```

$$y = \frac{1}{\operatorname{sen}x}$$



```
> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-1/cos(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función secante")
> |
```

$$y = \frac{1}{\operatorname{cos}x}$$

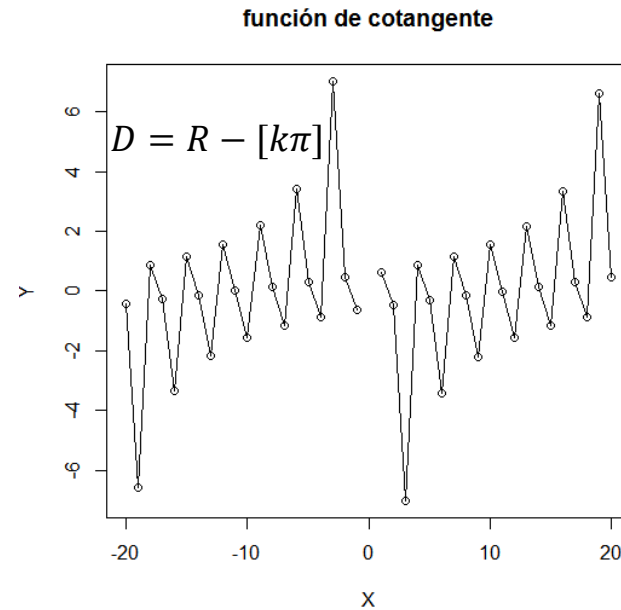


```

> desde<--20
> hasta<-20
> Y<-numeric(1+hasta-desde)
> n<-length(Y)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-1/tan(desde)
+ X[i] <-desde
+ desde=desde+1
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función de cotangente")
> |

```

$$y = \frac{1}{\operatorname{tg}x}$$



Funciones trigonométricas recíprocas

asin() • $x < 0 = 1$ arcoseno en el intervalo $-\pi/2$ y $\pi/2$ definida entre -1 y +1

acos() • $x > 0 = 1$ arcocoseno en el intervalo $-\pi/2$ y $\pi/2$ definida entre -1 y +1

atan() • arcotangente

atan2() • arcotangente en el intervalo entre $-3\pi/2$ y $3\pi/2$. **atan2(y = -2, x = 0.9)**

Funciones hiperbólicas

sinh() • seno hiperbolico **sinh(x = 2.45)**

cosh() • coseno hiperbolico **cosh(x = 2.45)**

tanh() • tangente hiperbolica **tanh(x = 2.45)**

asinh() • inverso seno hiperbolico **asinh(x = 2.45)**

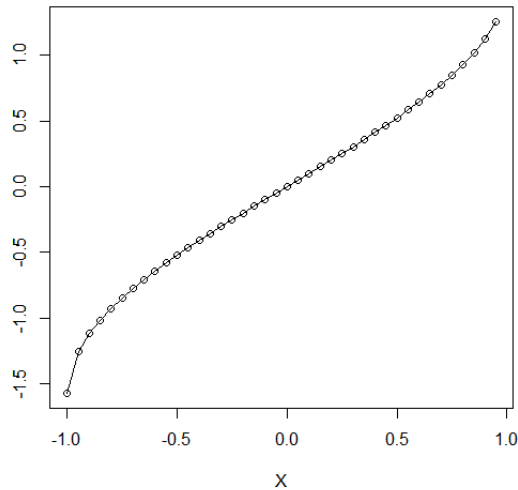
acosh() • inverso coseno hiperbolico **acosh(x = 2.45)**

atanh() • $x < 0 = 1$ inversa tangente hiperbolica **atanh(x = 0.45)**

FUNCIONES TRIGONOMÉTRICAS RECÍPROCAS

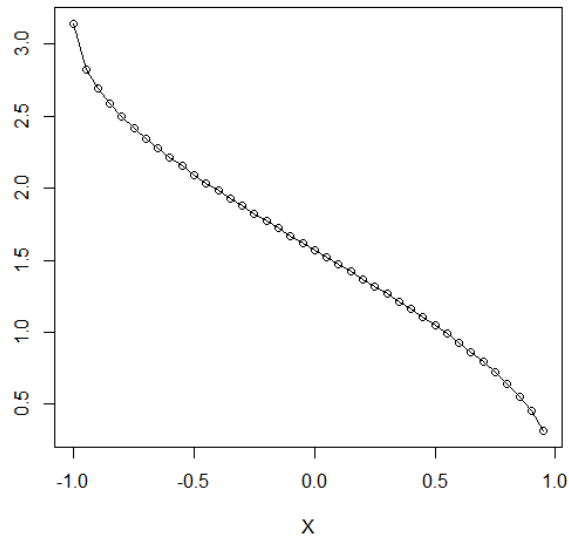
```
> desde<--1
> hasta<-1
> n=40 #iteracciones
> lapsus <-2/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-asin(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función arcoseno")
> |
```

función arcoseno



```
> desde<--1
> hasta<-1
> n=40 #iteracciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-acos(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función arcocoseno")
> acos(pi)
[1] NaN
Mensajes de aviso perdidos
In acos(pi) : Se han producido NaNs
> |
```

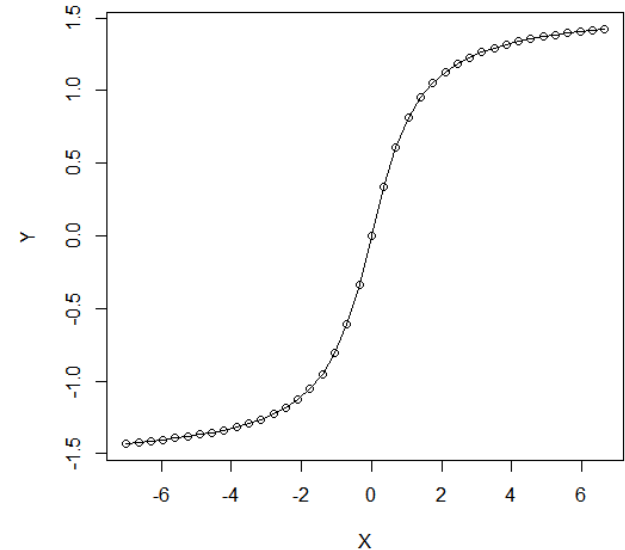
función arcocoseno



```
> desde<--7
> hasta<-7
> n=40 #iteracciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-atan(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función arcotangente")
> atan(20)
[1] 1.520838
> atan (80)
[1] 1.558297
> |
```

D=R

función arcotangente



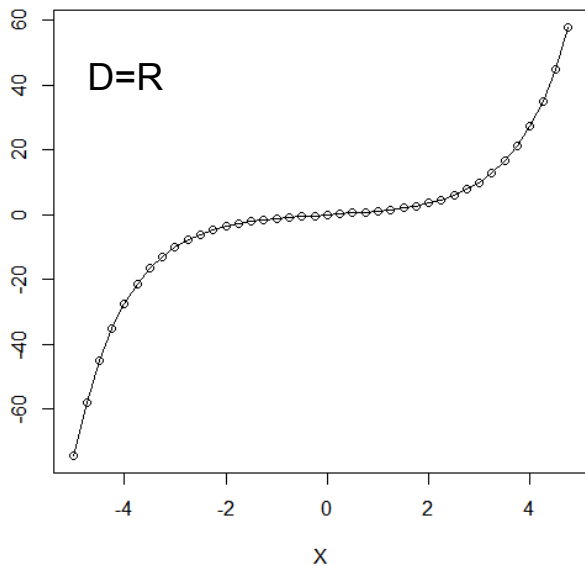
FUNCIONES HIPERBÓLICAS

```
> desde<--5
> hasta<-5
> n=40 #iteraciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-sinh(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función seno hiperbólico")
```

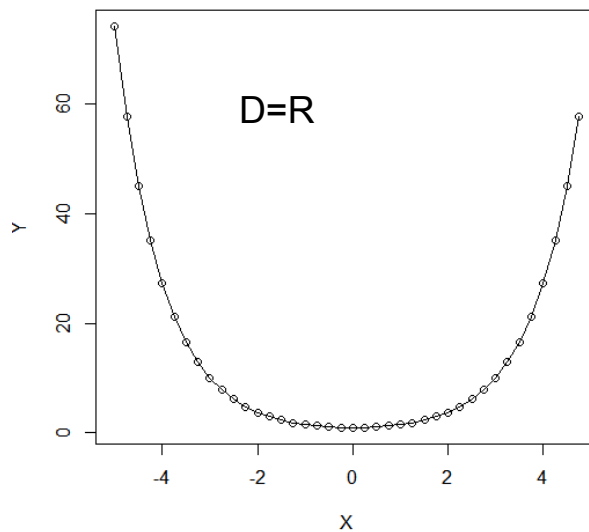
```
> desde<--5
> hasta<-5
> n=40 #iteraciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-cosh(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función coseno hiperbólico")
> |
```

```
> desde<--5
> hasta<-5
> n=40 #iteraciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-tanh(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="función tangente hiperbólica")
\ |
```

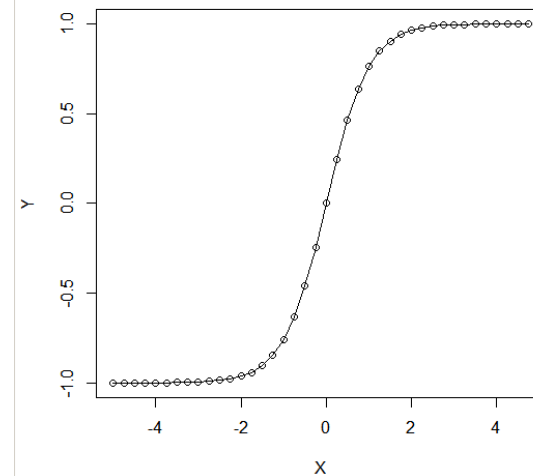
función seno hiperbólico



función coseno hiperbólico



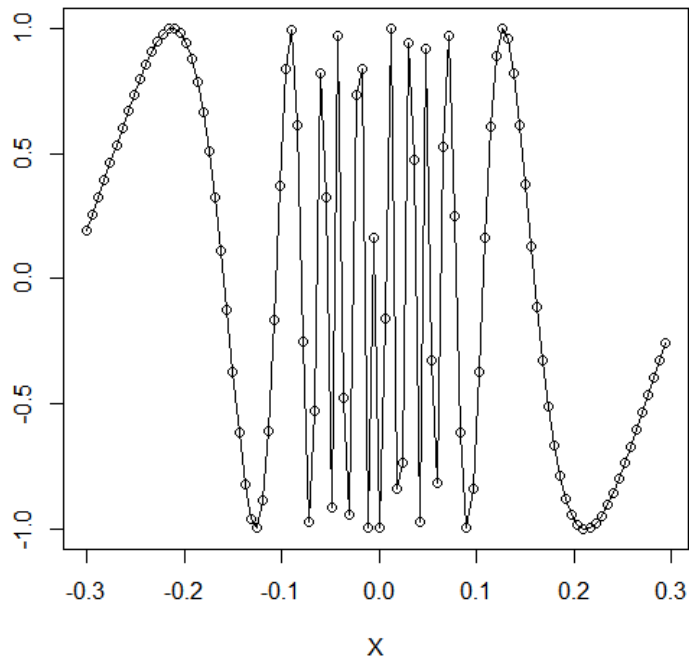
función tangente hiperbólica



DISCONTINUIDADES

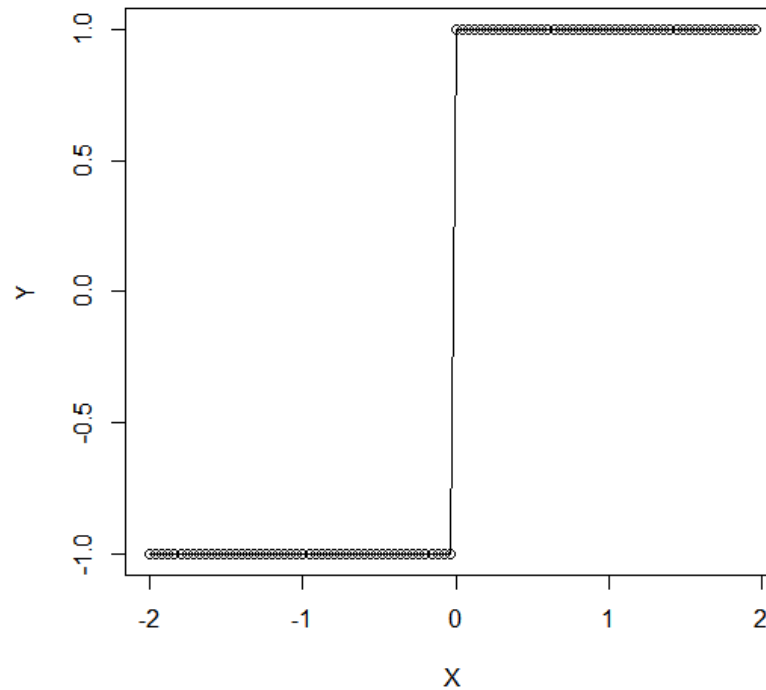
```
> desde<--0.3
> hasta<-0.3
> n=100 #iteracciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-sin(1/desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="Discontinuidad de 2ª especie")
< |
```

Discontinuidad de 2ª especie



```
> desde<--2
> hasta<-2
> n=100 #iteracciones
> lapsus <-(hasta-desde)/n
> Y<-numeric(n)
> X<-numeric(n)
> for (i in 1:n){
+ Y[i] <-desde/abs(desde)
+ X[i] <-desde
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="Discontinuidad de 1ª especie")
> |
```

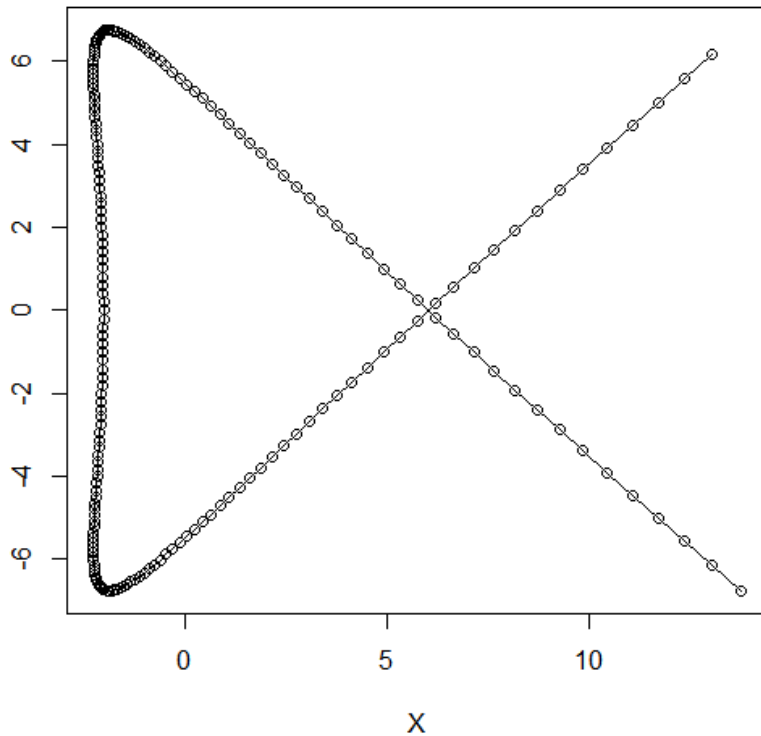
Discontinuidad de 1ª especie



CURVAS PARAMÉTRICAS

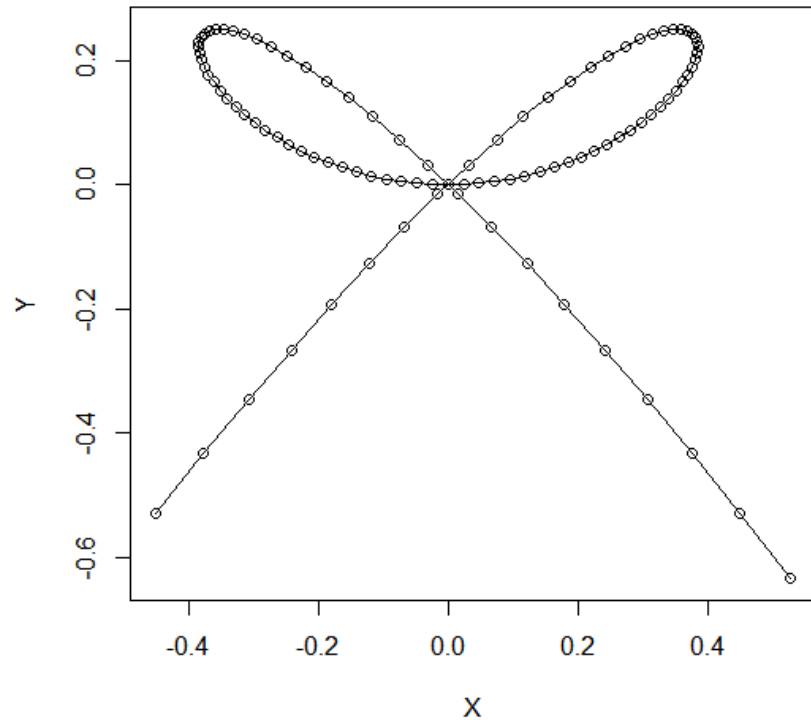
```
> desde<--3
> hasta<-3
> n=200 #iteracciones
> lapsus <- (hasta-desde)/n
> X<-numeric(n)
> Y<-numeric(n)
> for (i in 1:n){
+ X[i] <- ((desde^4)/4) - ((desde^2)/2) -2
+ Y[i] <- (desde^3) - ((27*desde)/4)
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="X=t^4/4-t^2/2 -2 Y=t^3-27.t/4")
> |
```

$$X=t^4/4-t^2/2-2 \quad Y=t^3-27.t/4$$



```
> desde<--1.2
> hasta<-1.2
> n=100 #iteracciones
> lapsus <- (hasta-desde)/n
> X<-numeric(n)
> Y<-numeric(n)
> for (i in 1:n){
+ X[i] <-desde-(desde^3)
+ Y[i] <- (desde^2) - (desde^4)
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="X=t-t^3 Y=t^2-t^4")
> |
```

$$X=t-t^3 \quad Y=t^2-t^4$$

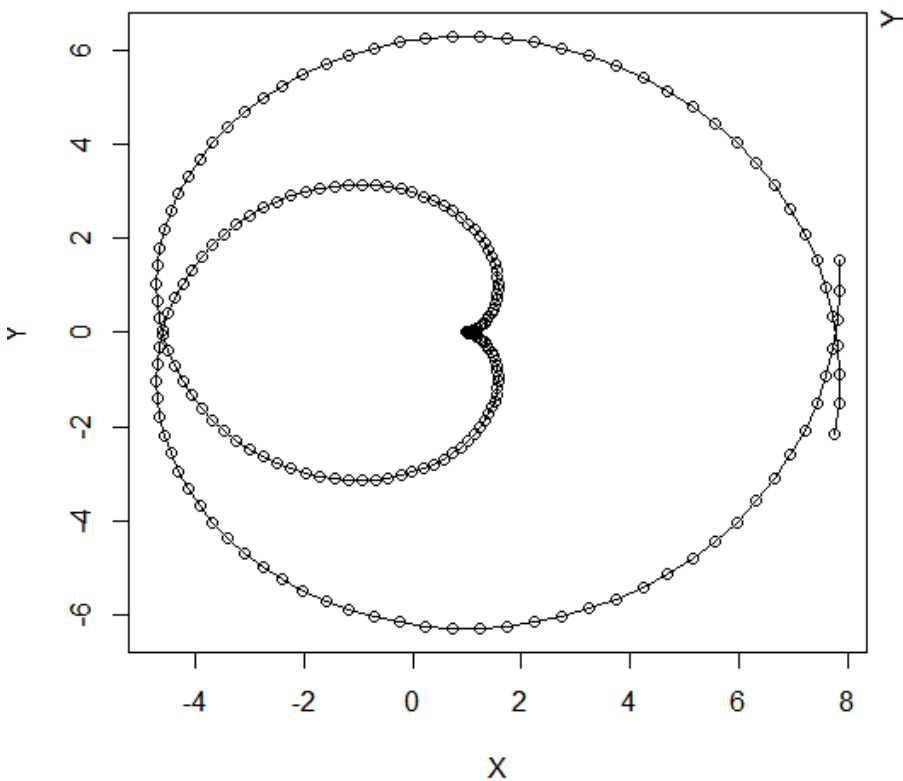


```

> desde<--8
> hasta<-8
> n=200 #iteracciones
> lapsus <-(hasta-desde)/n
> X<-numeric(n)
> Y<-numeric(n)
> for (i in 1:n){
+ X[i] <-cos(desde)+(desde*sin(desde))
+ Y[i] <-sin(desde)-(desde*cos(desde))
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="X=cos t+t sen t Y=sen t -t cos t")
.

```

X=cos t+t sen t Y=sen t -t cos t

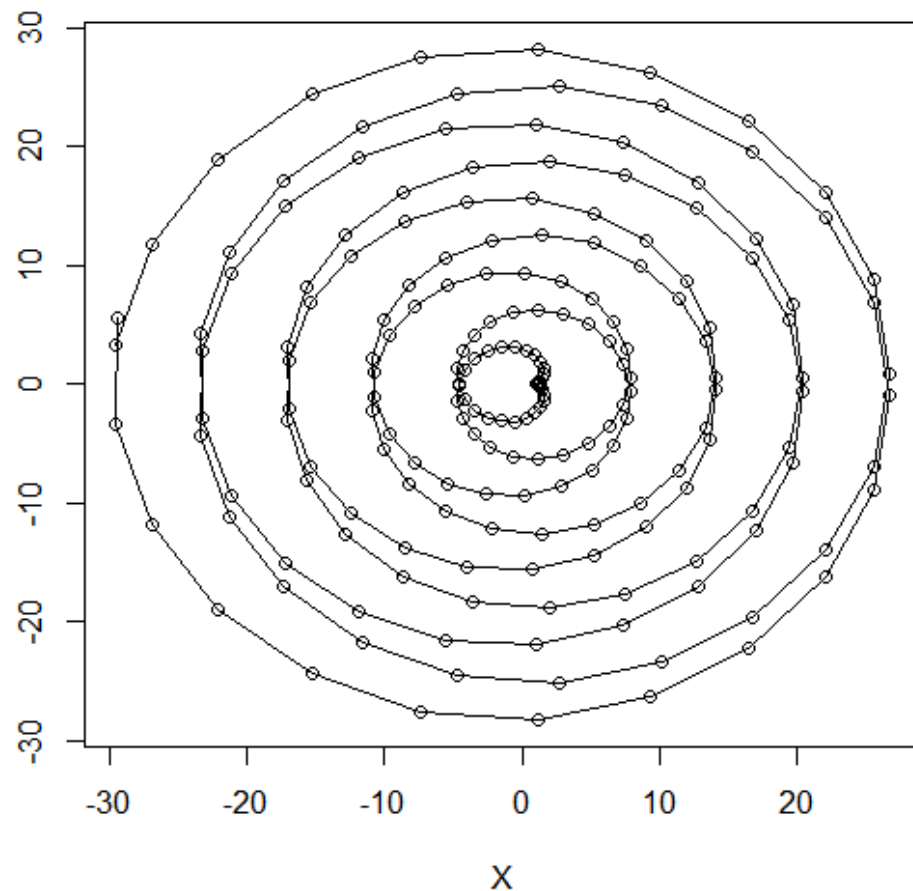


```

> desde<--30
> hasta<-30

```

X=cos t+t sen t Y=sen t -t cos t



```
> desde<--30
> hasta<-30
> n=200 #iteracciones
> lapsus <-(hasta-desde)/n
> X<-numeric(n)
> Y<-numeric(n)
> for (i in 1:n){
+ X[i] <-sin(5*desde)*cos(desde)
+ Y[i] <-sin(5*desde)*sin(4*desde)
+ desde=desde+lapsus
+ }
> plot (X,Y)
> lines (X,Y)
> title (main="X=sen 5t cos t Y=sen 5t sen 4t")
> |
```

X=sen 5t cos t Y=sen 5t sen 4t

