

FUNCIONES Y CONDICIONES

ÍNDICE

NUEVOS COMANDOS EMPLEADOS	2
ERRORES	3
EJERCICIOS	
EJERCICIO 1	4
EJERCICIO 2.....	6

NUEVOS COMANDOS EMPLEADOS:

FUNCIONES EN R:

Nombre de la función= `function`(argumento1,argumento2,argumentoN){
 expresión de la función
}

- `par("new true")` → para superponer curvas.
- `axes=FALSE` → para eliminar los ejes. Se utiliza dentro del plot.
- `legend(x="ubicación", c("función 1", "función 2", "función 3"), fill=c("color1", "color2", "color3"))` → para añadir leyenda. Lo escribimos después de plot de cada una de las funciones.
 - `x=""` → nos indica dónde se va a añadir la leyenda.
 - `c()` → nombres de las funciones.
 - `fill=c()` → el color de cada una de las funciones.
- `xx=seq(valor inicial, valor final, incremento)` . También se puede usar de la siguiente forma:
`xx=seq(valor inicial, valor final, length=cantidad de números que quieres)`.

ESTRUCTURAS CONDICIONALES:

- `If (Condición 1){` → condiciones
 - Proceso 1
 - }`else if (Condición 2){`
 - Proceso 2
 - }`else if (Condición n){`
 - Proceso n
 - }
- `While (condición){` → bucles condicionales
 - Proceso
 - }

ERRORES:

1. Siempre que se abra un paréntesis, corchete o llave se debe cerrar.

ERROR	<code>f= function(x){x^2*cos(x^2)}</code>
✓	<code>f= function(x){x^2*cos(x^2)}</code>

2. Cuando llamas a una función debes introducir tantos valores como valores de entrada tiene la función.

ERROR	<code>f=function(x,y){x^2*cos(y^2)}</code> <code>f(0.25)</code>
✓	<code>f=function(x,y){x^2*cos(y^2)}</code> <code>f(0.25,0.75)</code>

3. Cuando generes una leyenda (legend), coloca "" en los colores que acompañan a las etiquetas definidas con legend.

ERROR	<code>legend(x="top", c("Función f", "Función g", "Función h"), fill = c(green, blue, red))</code>
✓	<code>legend(x="top", c("Función f", "Función g", "Función h"), fill = c("green", "blue", "red"))</code>

4. En la representación de dos curvas en un mismo gráfico, utiliza la estructura:
`plot()`
`par(new=TRUE)`
`plot()`

ERROR	<code>plot(xx, f(xx), type="b", col="green", xlab="Abscisas", ylab="Mis funciones f, g y h")</code> <code>plot(xx, g(xx), type="b", col="blue", xlab="", ylab="", axes= "FALSE", pch=4)</code>
✓	<code>plot(xx, f(xx), type="b", col="green", xlab="Abscisas", ylab="Mis funciones f, g y h")</code> <code>par(new="true")</code> <code>plot(xx, g(xx), type="b", col="blue", xlab="", ylab="", axes= "FALSE", pch=4)</code>

5. Escribe todas las funciones al principio del programa.

EJERCICIOS:

EJERCICIO 1:

1. Define la función $f(x) = x^2 \cos(x^2)$ y obtén el valor $f(\pi/4)$.
2. Obtén un vector `xx` que tome 1001 valores en $[0,10]$ (utilizando el comando `seq` con «salto» 0.01).
3. Representa gráficamente la función f , tomando como abscisas los valores `xx`. Etiqueta el eje de abscisas con "Abscisa" y el eje de ordenadas con "mis funciones f, g, h". La gráfica irá en color verde.
4. Define la función $g(x) = \sin(x^2)e^{-x^2/10}$
5. Utiliza la instrucción: `par(new="true")` para superponer curvas.
6. Dibuja la función g en los puntos `xx` en color azul y empleando: `xlab=""`, `ylab=""`, `axes=FALSE` (para eliminar ejes), `pch=4` (para símbolos)
7. Idem a 5-6 con la función $h(x) = \sin(x^8)e^{-x}$ en rojo y usando `pch=18`
8. Añade a continuación la leyenda: `legend(x="top", c("Función f", "Función g", "Función h"), fill = c("green", "blue", "red"))`

RESOLUCIÓN (CONSOLA):

Define la función $f(x) = x^2 \cos(x^2)$ y obtén el valor de $f(\pi/4)$

```
f=function(x){x^2*cos(x^2)}  
f(pi/4)
```

Define la función $g(x) = \sin(x^2)e^{-x^2/10}$

```
g = function(x){sin(x^2)*exp(-x^2/10)}
```

Define la función $h(x) = \sin(x^8)e^{-x}$

```
h = function(x){sin(x^8)*exp(-x)}
```

Obtén el vector `xx` que tome 1001 valores en $[0,10]$ (utilizando el comando `seq` con "salto" 0.01)

```
xx= seq(0,10,0.01)  
xx
```

Representa gráficamente la función f , tomando como abscisas los valores de `xx`. Etiqueta el eje de abscisas con "Abscisa" y el eje de ordenadas con "mis funciones f, g, h". La gráfica irá en color verde

```
plot(xx, f(xx), type="b", col="green", xlab="Abscisas", ylab="Mis funciones f, g y h")
```

Utiliza la instrucción: `par(new="true")` para superponer curvas

```
par(new="true")
```

Dibuja la función g en los puntos `xx` en color azul y empleando: `xlab=""`, `ylab=""`, `axes=FALSE` (para eliminar los ejes), `pch=4` (para símbolos)

```
plot(xx, g(xx), type="b", col="blue", xlab="", ylab="", axes= "FALSE", pch=4)
```

Idem con la función $h(x) = \sin(x^8) e^{-x}$ en rojo y usando `pch=18`

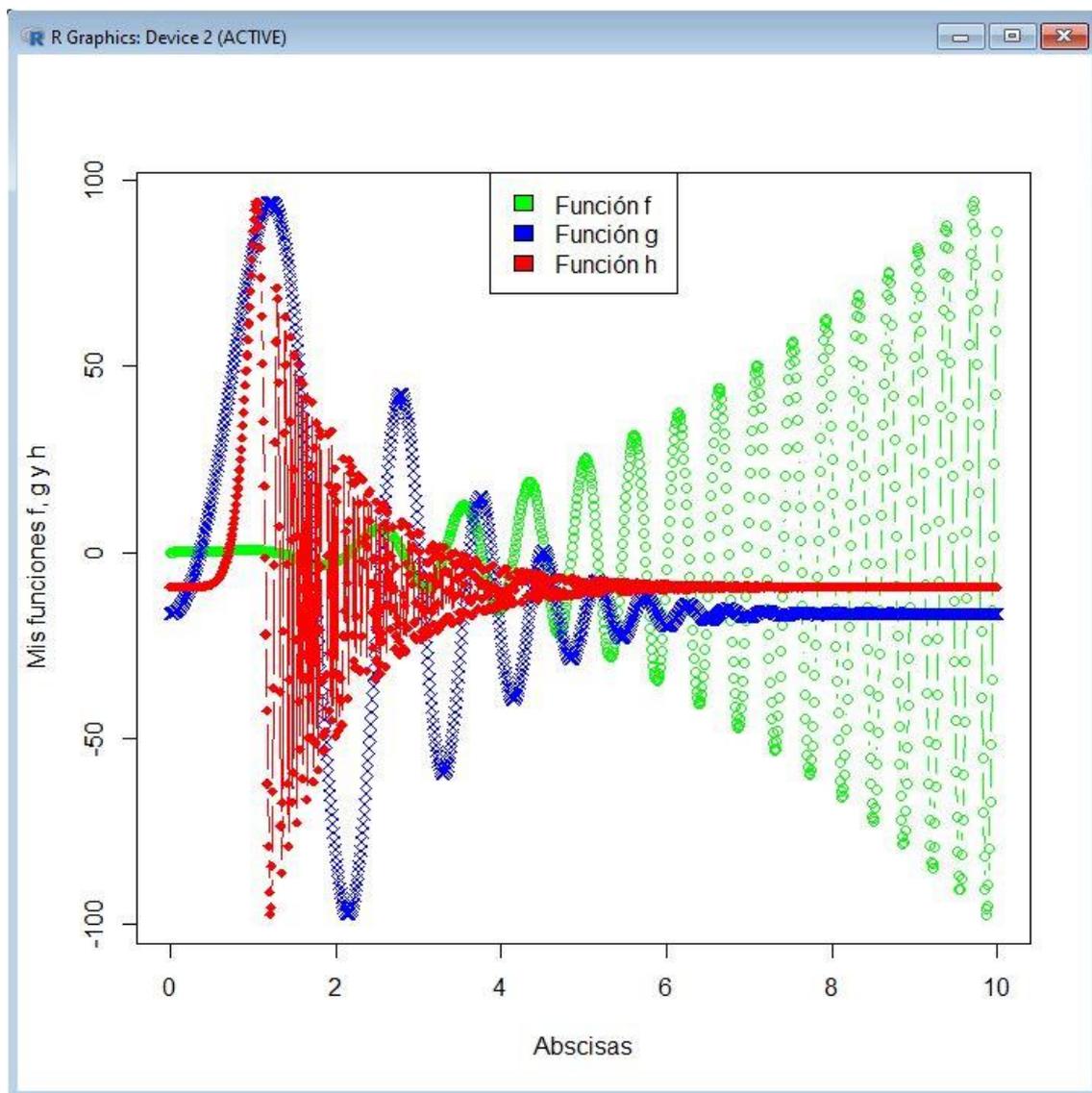
```
par(new="true")
```

```
plot(xx, h(xx), type="b", col="red", xlab="", ylab="", axes="FALSE", pch=18)
```

Añade la leyenda:

```
legend(x="top", c("Función f", "Función g", "Función h"), fill = c("green", "blue", "red"))
```

GRÁFICA 1:



EJERCICIO 2:

Se está estudiando la medicación a administrar ante una determinada infección.

Se sabe que la evolución de las células infectadas en función del tiempo sigue la ley

$$N(t) = 10\exp(t/20) (*)$$

donde N es el número de nuevas células infectadas y t es el tiempo.

Con el objeto de decidir el tipo de medicación se seguirá el siguiente criterio:

- Si $N < 5000$, enfermo recuperable sin medicación.
- Si $N \geq 5000$ y $N \leq 10000$, medicación moderada.
- Si $N > 20000$, medicación agresiva.
- En otro caso se trata de una situación intermedia.

El estudio se realiza durante una semana (168 horas) calculando el número de células cada media hora ($dt = 0.5$).

La función que determina el número de células infectadas en función del tiempo (*) se programará en una function de R.

Al final del proceso se realizará un gráfico tipo histograma representando el número de células infectadas en función del tiempo, representando en color verde el caso sin medicación, en color naranja el caso medicación moderada, en color rojo el caso medicación agresiva y en color azul el caso de situación intermedia.

Para realizar la representación gráfica se almacenarán los valores del tiempo en un vector xd y el número de células infectadas en un vector yd. Los colores se almacenarán en un vector llamado color.

RESOLUCIÓN (CONSOLA):

```
N=function(t){10*exp(t/20)}
i=1
xd=0
yd=0
color=0
t=0
tfin=168
dt=0.5
while (t<=tfin){
  xd[i]=t
  yd[i]=N(t)
  if(N(t)<5000){
    color[i]="green"
  }else if (N(t)>=5000 & N(t)<10000){
    color[i]="orange"
  }else if (N(t)>20000){
    color[i]="red"
  }
  i=i+1
  t=t+dt
}
```

```

    }else{
    color[i]="blue"
    }
t=t+dt
i=i+1
}

```

```

plot(xd, yd, type="h", col=color, xlab="Tiempo (horas)", ylab="Células infectadas",
main="Medicación")
legend (x="top", c("Sin medicación","Medicación moderada","Medicación intermedia","Medicación
intensa"), fill=c("green","orange","blue","red"))

```

GRÁFICA 2:

