

Errores Comunes o Posibles - Práctica 4

Ejercicio 1



Funciones en R

```
Nombre de la función <-  
function(argumento1,argumento2,...){  
  expresión de la función  
}
```

Ejercicio:

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 $g(x) = \sin(x^2)e^{-\frac{x^2}{10}}$
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"))

Solución:

```
#USO DE FUNCIONES Y REPRESENTACIÓN GRÁFICA  
# colocamos todas las funciones al inicio del programa  
#uso <- o =, en vez de x puedo poner lo que quiera  
f<-function(x){x^2*cos(x^2)}  
g<-function(x){sin(x^2)*exp(-x^2/10)}  
h<-function(x){sin(x^8)*exp(-x)}  
  
# f en pi/4  
f(pi/4)  
  
# generar vector xx que tome 1001 valores en [0,10] utilizando el comando  
seq con salto 0.01  
xx<-seq(0,10,0.01)  
  
# particularizar la función f en todos los valores de xx(a cada valor le aplica f)  
f(xx)  
  
#representaciones gráficas:  
plot(xx,f(xx),xlab='abscisa',ylab='mis funciones f,g,h',col='green')  
par(new='true')  
plot(xx,g(xx),xlab="",ylab="",col='blue',axes=FALSE,pch=4)
```

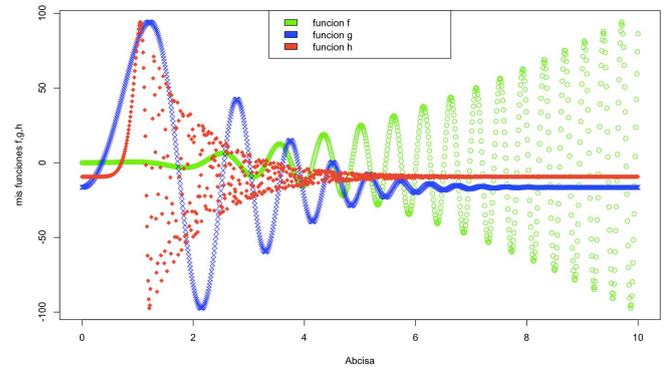
```

par(new='true')
plot(xx,h(xx),xlab="",ylab="",col='red',axes=FALSE,pch=18)
#leyenda
legend(x='top',c('f','g','h'),fill=c('green','blue','red'))

```

Errores más comunes:

- Poner verdadero en vez de true y falso en vez de false
- Olvidarse que el número e se define como exp(), y poner sen en vez de sin
- No poner ^ para definir un exponente
- Poner un igual (=) después de legend
- Olvidar las comillas



Ejercicio 2



Ejercicio 1. Medicación

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

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

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 ($dt = 0.5$).

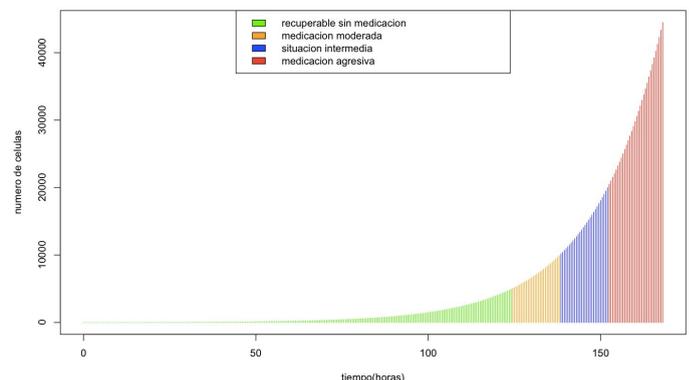
La función que determina el número de células infectadas en función del tiempo (*) se programará en una función de R.

Al final del proceso se realizará un gráfico tipo histograma representando el número de células infectadas en el 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.

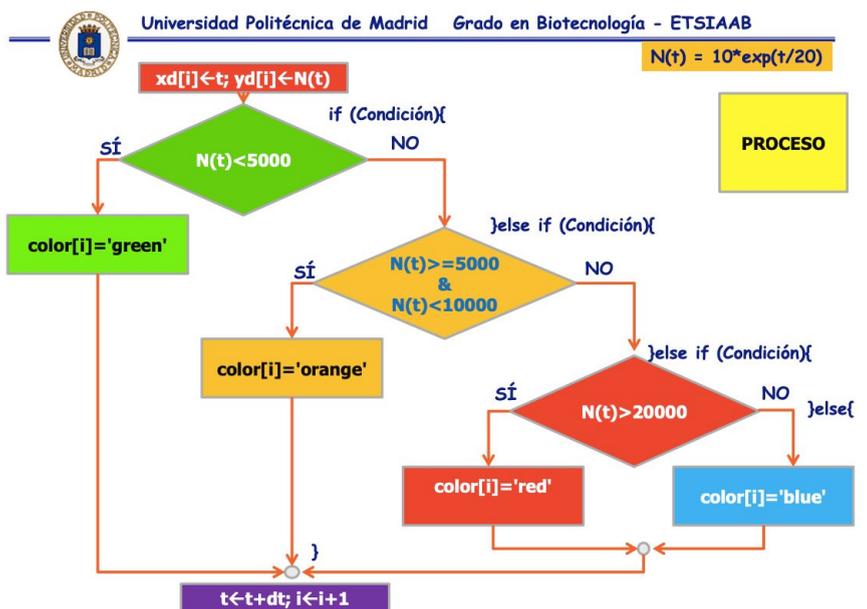
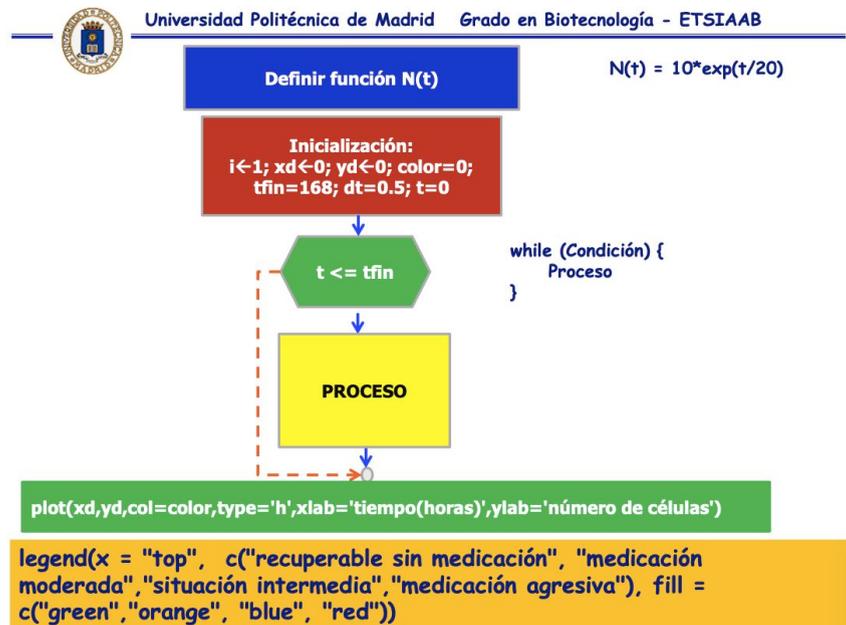
Para realizar la representación gráfica se almacenarán los valores del tiempo en un vector xd y el número de células infectadas en un vector yd. Los colores se almacenarán en un vector llamado color

Solución:

```
N=function(t)(10*exp(t/20))
i=1; xd=0; yd=0; color=0; tfin=168; dt=0.5; t=0
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'
  }else{
    color [i]='blue'
  }
  t=t+dt; i=i+1
}
plot(xd,yd,col=color, type='h',xlab='tiempo(horas)',ylab='numero de celulas')
legend(x='top', c('recuperable sin medicacion','medicacion moderada', 'situacion
intermedia','medicacion agresiva'), fill=c('green', 'orange', 'blue', 'red'))
```



Además, el profesor nos proporciona unos organigramas para entender mejor el funcionamiento de los bucles:



Errores más comunes:

- Para darle color a la función, no utilizar `color=' '`. La forma correcta es `col=' '`.
- En el comando `xlab=' '`, no se debe olvidar poner el =

- Cuando se realizan bucles condicionales, ponemos **else{** en último lugar y sin condición, el resto como **if(Condicion1){** o **else if(Condicion2){**, siempre llevan condición