

# PRÁCTICA 4

En este documento hay una guía para entender cómo resolver el ejercicio de la práctica 4:

## ESTRUCTURAS CONDICIONALES

### Enunciado:

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 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`.

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".

Es fundamental entender lo que nos están pidiendo:

Existe una función,  $N(t)$ , que dependiendo de si los valores que devuelve pasan o no ciertos límites la situación se considera con mayor o menor riesgo.

Para estudiarla, tomaremos múltiples valores a lo largo del intervalo  $[0, 168]$ , con una separación entre ellos de  $dt=0.5$ . Con cada uno de esos valores, que en el bucle while los guardamos en el vector "xd", obtendremos un valor de la función que guardaremos en el vector "yd".

Una vez hayamos obtenido todos los valores de la función  $N(t)$  en el intervalo (guardados todos en el vector yd), creamos una gráfica a modo de histograma (de barras verticales) y cambiamos los colores según los límites que nos establece el enunciado mediante condicionales.

Una vez entendido el procedimiento, toca crear el código paso a paso:

### **PASO 1: Crear la función $N(t)$**

Para ello escribimos  $N = \text{function}(t) \{ \}$

Esto significa que la función, llamada N, tiene como variable la letra t. Dentro de las llaves introduciremos las operaciones de la función.

```
N = function(t) {10*exp(t/20)}
```

### **PASO 2: Crear las variables**

Vamos a necesitar todas estas variables a lo largo del programa:

- $i$  ; Nos sirve para poder contar cuántas veces hemos realizado el proceso y así poder introducir los valores en las ranuras correspondientes de cada vector
- $xd$  ; Es un vector que almacenará todos los valores de abscisas
- $yd$  ; Es un vector que almacenará todos los valores de ordenadas
- $color$  ; Es un vector que almacenará el color que deberá tener la barra para cada valor de la función
- $tfin$  ; Tendrá el valor del último valor a tomar en la función. Nos dice el enunciado que es igual a 168
- $dt$  ; Tendrá el valor de la separación entre valores de abscisas a tomar en la función
- $t$  ; Nos sirve para dar valores de abscisas a  $xd$ . Esta variable empezará con valor 0, irá subiendo cada ciclo lo que indique en  $dt$  (que nos dice el enunciado que es 0.5) y así hasta llegar al valor de  $tfin$  (que nos dice el enunciado que es 168)

```
i=1
xd=0
yd=0
color=0
tfin=168
dt=0.5
t=0
```

### **PASO 3: Crear el bucle while**

Vamos a crear un bucle while en el que, empezando la variable  $t$  con valor 0, se repetirá lo que haya en su interior hasta que  $t$  sea igual a  $dt$  (es decir, 168). Además,  $t$  variará en cada ciclo lo que se indique en  $dt$  (que es 0.5). Así pues:

```
while (t<=tfin) {  
  #EL RESTO DEL CÓDIGO DEL BUCLE  
  t=t+dt  
  i=i+1  
}
```

Esto dice: mientras  $t$  sea menor o igual que  $tfin$ , se realizará lo siguiente:

— Lo que queramos hacer en el bucle —

Y al final cambiar el valor de  $t$  (y el de  $i$  para llevar la cuenta)

### **PASO 4: Completar el bucle while**

Queda por programar lo que antes hemos dejado en blanco en el bucle. En este hueco tenemos que hacer 2 cosas:

- Añadir a  $xd$  el valor de este ciclo, y a  $yd$  el suyo sabiendo que es igual a  $N(xd)$
- Añadir al vector *color* el color (se indica con el nombre de ese color entre comillas. Por ejemplo: "green" o "blue") siguiendo las normas que nos impone el enunciado

Para la segunda tarea necesitaremos condicionales: el *if* del principio mira si el valor de la función en ese punto está por debajo de 5000. Si está por debajo el color es verde y todos los otros condicionales se ignoran, pero si no lo está pasa al siguiente condicional.

Al llegar a los *else if*, cada uno de nuevo mira su condición: si se cumple se termina el condicional y se hace lo que ponga entre llaves, pero si no se cumple se mira la siguiente condición.

Es importante destacar que utilizamos los comandos *else/else if*, para que en caso de no cumplirse la condición anterior, se ejecuten las operaciones alternativas. En caso de que utilizásemos solo el comando *if*, las operaciones de se ejecutarían de manera continuada, y no aparecerían todos los valores en la gráfica.

Finalmente al llegar al *else*, el código entre llaves se ejecuta sí o sí, no hay condición. Si se ha llegado al *else*, significa que ninguna de las condiciones anteriores se ha cumplido, y en este caso es porque es una situación intermedia

```
while (t<=tfin) {  
  xd[i]=t  
  yd[i]=N(t)  
  if (N(t)<5000) {color[i]="green"}  
  else if (N(t)<10000) {color[i]="orange"}  
  else if (N(t)>20000) {color[i]="red"}  
  else {color[i]="blue"}  
  t=t+dt  
  i=i+1  
}
```

Aunque la opción propuesta es la que se muestra en la imagen superior, podríamos encontrar otras formas de hacerlo, de manera que los valores dados a los condicionales *e/else if*, podrían variar, pero obtendríamos el mismo resultado, como se muestra en la imagen inferior.

```
while (t<=tfin) {  
    xd[i]=t  
    yd[i]=N(t)  
    if (N(t)<5000) {color[i]="green"}  
    else if (N(t)<10000) {color[i]="orange"}  
    else if (N(t)<20000) {color[i]= "blue"}  
    else {color[i]="red"}  
    t=t+dt  
    i=i+1  
}
```

### **PASO 5: Crear la gráfica con un *plot()***

Escribimos *plot()* y en el interior del paréntesis ponemos toda la información: Primero las abscisas, luego las ordenadas, y seguidamente todo el resto de datos (sin necesidad de ordenamiento para estos).

Para que sea tipo histograma, ponemos *type="h"*.

En caso de que no nos indiquen el tipo de grafica que quieren, podremos optar por diferentes formatos, por ejemplo; *type "p"* mostrará una gráfica de puntos, *type "l"*, una gráfica de líneas, y *type "b"* (both), una gráfica de puntos unidos por líneas.

Es importante añadir el color con *col=color*, indicándole que el color es el que reside en la posición correspondiente del vector color. También ponemos título a los ejes con *xlab=""* e *ylab=""* (yendo entre las comillas el título de cada eje).

```
plot(xd,yd,col=color,type='h',xlab='Tiempo (horas)',ylab='Número  
de células')
```

### **PASO 6: Crear la leyenda**

Para rematar la labor, crearemos una leyenda indicando qué colores significan qué situaciones en la medicación.

Para ello, en una nueva línea (no en el *plot()*) escribiremos *legend()*, con todos los datos entre los paréntesis. Nosotros hemos puesto:

- *x = "top"*; Esto le indica al programa que queremos que coloque la leyenda arriba
- *c(Recuperable sin...)*; Esto es un vector inventado sobre la marcha que tiene como componentes los cuatro estados que puede tener la gráfica. Los acompañaremos

con el color de cada uno para informar al usuario de sus significados en la gráfica

- `fill = c("green", ...)` ; Esto es un vector también inventado sobre la marcha. Tiene 4 colores, y el orden es el mismo que el del vector de los estados de medicación. El programa los relacionará y asociará cada estado con el color que le toca

```
legend(x = "top", c("Recuperable sin medicación", "Medicación moderada", "Situación intermedia", "Medicación agresiva"), fill = c("green", "orange", "blue", "red"))
```

El resultado final de la gráfica sería el siguiente:

