

EJERCICIOS GUIÓN 3

En el guión de la práctica 4 de fundamentos de programación se tratan las estructuras condicionales.

Es el caso del comando “if”

El “if” es un comando de bifurcación por una condición. Al llegar al if el programa se pregunta si se da la condición que hay en el if. Si se cumple el programa hace una operación (**puede ser que no haga nada**) y si no se cumple hará otra cosa (**puede ser que no haga nada tampoco**).

Es importante saber que no es un bucle ya que una vez que pasas por el comando no se vuelve a repetir (**a no ser que esté en otro bucle**)

Dentro del comando “if” hay varias variantes:

- Si queremos que después de pasar por una condición, el programa pase por otra, utilizaremos “else if”
- Si queremos incluir una última acabaremos el comando con un “else”

```
if(N(t)<5000){
    color[i]='green'
}else if(N(t)>=5000 & N(t)<10000){
    color[i]='yellow'
}else if(N(t)>20000){
    color[i]='red'
}else{
    color[i]='blue'
}
```

EJERCICIO DE LA MEDICACIÓN

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**

Importante: LEER BIEN EL ENUNCIADO

COMPRENSIÓN

-Nos dan la función del número de células infectadas en función del tiempo y nos piden que hagamos un estudio de cuántas células infectadas hay cada 30 min (**0,5 h**) durante 168h.

-A partir de estos datos tendremos que representar la función de células infectadas con los datos que nos proporciona el estudio teniendo en cuenta los valores referencia que nos da el enunciado y el color que deben tener

-Como nos dice el enunciado, los datos serán guardados en unos vectores para su representación, es decir, un vector "**xd**" para los tiempos en los que se han tomado los datos, y un vector "**yd**" para los valores que toma la función en esos tiempos

-Aparte nos dice que los colores los guardamos en un vector "**color**"

-Nosotros recomendamos que tengáis antes de empezar la práctica una visión general de los comandos que váis a utilizar en el programa:

- un "**function**" para la función de células infectadas
- un "**while**" para ir generando los vectores "**yd**" (**células infectadas**) ya que lo que estaremos haciendo es guardar cada valor de la función "**N**" en el vector "**yd**" mientras el tiempo esté dentro del de nuestro rango de estudio (**habrá que ir aumentando este tiempo en 0,5h**)
- un "**if**" dentro del bucle "**while**" para ir almacenando los colores (**en el vector "color"**) correspondientes a cada valor de la función que estamos guardando en el vector "**yd**".
- un "**plot**" para generar la gráfica

REALIZACIÓN

1. Primero utilizamos `rm(list=ls(all=TRUE))` para borrar todas las variables guardadas en ese momento (**recomendación del grupo**)
2. Vamos a programar la función $N(t)$. Para ello utilizaremos el comando `function`. ¿Qué nombre le ponemos a la función? -> `"N"` y ¿de qué depende `"N"`? Del tiempo `"t"`. El comando quedaría así

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

```
} (recomendamos poner las llaves escalonadas)
```

3. Una vez definida la función nos tenemos que hacer la pregunta ¿Qué datos necesitamos? (**se podría hacer antes de la función**)
 - Un dato de tiempo máximo `"tiempomax=168"`
 - Un dato de tiempo inicial `"t=0"`
 - Un dato de la diferencia de tiempo entre toma y toma de datos `"dt=0,5"`
 - Inicializamos valores de los vectores (**necesario para el bucle while ya que si no da error**): `"xd=0"` ; `"yd=0"` ; `"color=0"`
 - Por último utilizaremos una variable `"i"` que será la encargada de ir metiendo los datos del bucle `"while"` en los vectores `"xd"` ; `"yd"` ; `"color"` -> `"i=1"`
4. Empezamos con el bucle `"while"`: Necesitamos que la función vaya metiendo valores en los vectores MIENTRAS el tiempo esté entre el tiempo inicial y el tiempo máximo. Por lo que la condición del bucle será `"while(t<=tiempomax){"`

Lo primero dentro del bucle será generar los vectores `"xd"` ; `"yd"`.

Los valores del vector `"xd"` serán simplemente los valores de tiempo `"t"` de cada ejecución del bucle -> `"xd[i]=t"`

Los valores del vector `"yd"` serán los valores de la función `"N(t)"` en cada unidad de tiempo -> `"yd[i]=N(t)"`

5. Dentro del bucle `"while"` introduciremos el comando `"if"` para generar el vector `"color"`. En este caso tenemos que traducir el enunciado: Si $N(t)$ es menor que 5000 el color será rojo. Traducimos: `if(N(t)<5000){`

```
    color[i]="green"
```

```
}
```

Seguimos traduciendo (utilizamos `"else if"`): Si $N(t)$ está entre 5000 y 10000 el color será naranja

```
    }else if(N(t)>=5000 & N(t)<=10000){
```

```
        color[i]="orange"
```

```
}
```

Continuamos traduciendo: Si $N(t)$ es mayor que 20000 el color será rojo:

```
    } else if(N(t)>20000){  
        color[i]="red"  
    }
```

Acabamos de traducir (utilizamos “else” solo para acabar): Si $N(t)$ no está entre estos valores será azul:

```
    }else{  
        color[i]="blue"  
    }
```

6. Por último tendremos que actualizar dentro del bucle al final las variables “i” ; “t”. “t” aumentará su “dt” y “i” solo 1
“t=t+dt” ; “i=i+1”

7. CERRAMOS BUCLE WHILE

8. Ahora generamos la gráfica con los vectores que hemos calculado: “xd” ; “yd” ; “color” utilizando el comando “plot”

- Eje x: el vector “xd”
- Eje y: el vector “yd”
- Color: el vector “color”
- Tipo de gráfico: histograma -> **type='h'**
- Nombre eje x: tiempo(horas) -> **xlab="tiempo(horas)"**
- Nombre eje y: número de células -> **ylab="número de células"**

El comando quedaría así:

```
“plot(xd,yd,col=color,type='h',xlab='tiempo(horas)',ylab='número de células)’”
```

9. Por último necesitamos incluir la leyenda que nos aclare los datos. Utilizaremos el comando “legend”.

- Lugar de la leyenda: arriba “x=top”
- Nombres que aparecerán en la leyenda: **c("recuperable sin medicación", "medicación moderada","situación intermedia","medicación agresiva")**
- Colores que aparecerán en la leyenda: **fill=c("green","yellow", "blue", "red")**

El comando quedaría así:

```
“legend(x = "top", c("recuperable sin medicación", "medicación moderada","situación intermedia","medicación agresiva"), fill =c("green","yellow", "blue", "red"))”
```

EL COMANDO QUEDARÍA ASÍ

```
rm(list=ls(all=TRUE))
N=function(t){
  (10)*exp(t/20)
}
xd=0
yd=0
t=0
tiempomax=168
diferencialt=0.5
color=0
i=1
while(t<=tiempomax){
  xd[i]=t
  yd[i]=N(t)
  if(N(t)<5000){
    color[i]='green'
  }else if(N(t)>=5000 & N(t)<10000){
    color[i]='yellow'
  }else if(N(t)>20000){
    color[i]='red'
  }else{
    color[i]='blue'
  }
  t=t+diferencialt
  i=i+1
}
plot(xd,yd,col=color,type='h',xlab='tiempo(horas)',ylab='número de células')
legend(x = "top", c("recuperable sin medicación", "medicación
moderada","situación intermedia","medicación agresiva"), fill =
c("green","yellow", "blue", "red"))
```

