

Resolución de la práctica 4 de R

Esta práctica consta de un solo ejercicio en el que se trabajará la programación y representación de funciones. El enunciado de dicho ejercicio es:

Se conoce la producción de bioetanol que se obtiene en una planta de biocombustibles en función del tiempo. Los valores de la concentración (g/l) están almacenados en el vector: B(10,20,35.33,35.5) y los instantes de tiempo (horas) en el vector s(1,3.5,5,10).

SE PIDE: Realizar un script llamado Bioetanol.R para estimar, mediante interpolación de Lagrange, la concentración de bioetanol en los instantes $t = 2$ y $t = 6.5$. Para ello, se empleará la siguiente expresión del polinomio interpolador de Lagrange:

$$p = \sum_{i=1}^n B_i L_i$$

Los pasos a seguir son los siguientes:

- 1. Programar la función Polbase para obtener las funciones de base la interpolación de Lagrange, que están dadas por la expresión:**

$$L_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{t - s_j}{s_i - s_j}, (i = 1, \dots, n)$$

$i \neq j$

Para llevar a cabo este ejercicio, vamos a emplear las funciones de R (o 'functions'). Estas, son una operación o una serie de operaciones realizadas a partir de unos datos de entrada, con los que se obtiene otros datos de salida. Para introducirlas en el programa, no es necesario indicar los datos antes, sino solamente indicar las variables de entrada, los procesos que se llevan a cabo y las variables de salida. Los datos de salida pierden el nombre que tenían en la función por lo que conviene darles uno (Paco=A(s,t,n)). Sigue la siguiente estructura:

```
A(el nombre de la función)=function(s,t,n (las variables que utiliza)){  
  ... (los procesos que queremos que la función lleve a cabo)  
  return(L) (la variable que queramos que salga)  
} (fin de la función)
```

Quedando:

```
A=function(s,t,n){  
  ...  
  Return(L)  
}
```

Esta función se puede emplear, más tarde, en cualquier parte del código del programa, siempre que se apliquen unos datos de entrada y otros de salida.

B(el nombre de una variable nueva)=A(el nombre de la función)(x,y,z)(los datos que queremos que use en el orden de las variables que queremos que sustituyan)

Quedando:

$B=A(x,y,z)$

La función de este ejercicio consiste en un productorio con dos variables (i,j), por lo que deberemos incluir los correspondientes bucles 'for'. También hay que tener en cuenta la condición de que $j \neq i$ y añadir una sentencia 'if'. Todas estas estructuras deberán estar incluidas entre los corchetes de la función 'Polbase'.

Tampoco podemos olvidar inicializar $L=0$.

```
Polbase=function(s,t,n){
    L=0
    for (i in 1:n) {
        L[i]=1
        for (j in 1:n) {
            if (i != j){
                L[i]=L[i]*(t-s[j])/(s[i]-s[j])
            } #cierre if
        } #cierre j
    } #cierre i
return(L) #especifica que valor quieres que salga.
} #cierre función
```

2. Programar la función 'PolInterp' para obtener el polinomio interpolador en el punto 't'.

$$p = \sum_{i=1}^n B_i L_i$$

En este caso, programar la función es mucho más sencillo ya que es un sumatorio con una sola variable (i), por lo que solo tendremos que incluir un bucle.

```
PolInterp=function(L, B, n){
  p=0
  for (i in 1:n) {
    p=p+(B[i]*L[i])
  }
  return(p)
}
```

3. Introducimos los datos.

Introducimos los datos después de las funciones para demostrar que estas no dependen de los datos, pueden ser modificados.

```
b=c(10,20,35.33,35.5)
```

```
s=c(1,3.5,5,10)
```

```
n=length(s)
```

```
tt=c(2,6.5)
```

A continuación, queremos calcular y dibujar conjuntamente los puntos y el polinomio interpolador.

```
Lbase1=Polbase(s,tt[1],n) #Polinomio de base 1 (L1)
```

```
pol1=PolInterp(b,Lbase1,n) #Polinomio interpolador de Lagrange 1 (B1*L1)
```

```
Lbase2=Polbase(s,tt[2],n) #Polinomio de base 2 (L2)
```

```
pol2=PolInterp(b,Lbase2,n) #Polinomio interpolador de Lagrange 2 (B2*L2)
```

Representación gráfica

```
x=seq(s[1],s[n], length=1001) #Calculamos 1001 puntos equidistantes para
representar la función
f=0
for (k in 1:1001){ #Sustituimos los valores equidistantes en las funciones
  lb=polbase(s,x[k],n)
  f[k]=polinterp(b,lb,n)
}
```

#Para los gráficos:

```
plot(x,f, type='l', lwd=3, col='green', ylim=c(0,60),
ylab='bioetanol(g/L)', xlab='tiempo (h)')
```

```
par(new='true') #para representar dos gráficos (los puntos y el polinomio
interpolador) a la vez
```

```
plot(s,b, ylim=c(0,60), xlab='', ylab='', main='Producción de bioetanol')
```

La gráfica resultante debería ser:

