PRÁCTICA 6 y 7 : Ejercicios sobre todo lo aprendido.

1. Desarrollo del ejercicio 1. Polinomio Interpolador de Lagrange

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 emplea la siguiente expresión del polinomio interpolador de Lagrange.

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

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

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

i != j

```
> Polbase=function(t,s,n){
+   L=c(0) #Necesario porque R lo exige
+   for(i in 1:n){
+     L[i]=1 #Necesario por el productorio
+     for(j in 1:n){
+       if(i!=j){
+         L[i]=L[i]*((t-s[j])/(s[i]-s[j]))
+       }
+     }
+   }
+   return(L)
+ }
```

2. Programar la función PolInterp para obtener el polinomio interpolador en el punto t. Dicha función recibirá como argumentos de entrada: • un vector B (que contiene los valores de la función que se interpola), • el vector L (que contiene los polinomios de base de Lagrange evaluados en el punto t) • la variable n (número de puntos del soporte de interpolación)

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

3. Introducir los datos y ejecutar las funciones programadas. Al ejecutar, se almacenarán las funciones de base en el vector L y el polinomio interpolador en la variable p.

```
> B=c(10,20,35.33,35.5)
> s=c(1,3.5,5,10)
> n=length(B)
```

```
> t=2
> L=Polbase(t,s,n)
> L
[1]  0.40000000  0.98461538 -0.40000000  0.01538462
> p=PolInterp(B,L,n)
> p
[1] 10.10646
>
> t=6.5
> L=Polbase(t,s,n)
> L
[1]  0.17500000 -1.18461538  1.92500000  0.08461538
> p=PolInterp(B,L,n)
> p
[1] 49.07179
```

4. Obtener 1001 abscisas equidistantes en el intervalo [s[1],s[n]] y almacenarlas en un vector x. (Calcularemos el valor del polinomio interpolador en cada abscisa para dibujar)

```
> x=seq(s[1],s[n], length=1001)
```

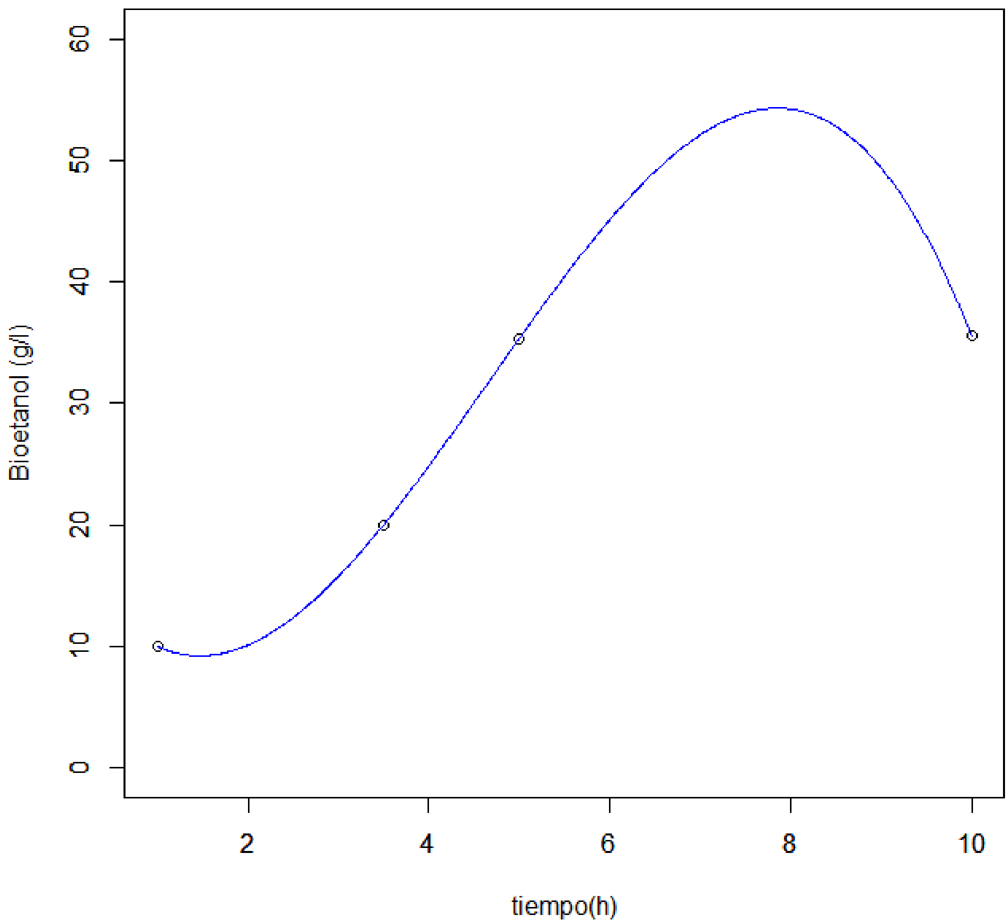
5. Llamamos 1001 veces a las funciones Polbase y PolInterp. La salida de PolInterp se guardará en un vector f[k], k=1,…,1001.

```
> f=c(0)
> for(k in 1:1001){
+   L=Polbase(x[k],s,n)
+   f[k]=PolInterp(B,L,n)
+ }
```

no tiene sentido meter todos los valores desde 0 hasta 1001, es por eso que creamos un bucle que recorra todos los valores uno a uno.

6. Dibujamos la gráfica

```
> plot(s,B,xlim=c(s[1],s[n]), ylim=c(0,60),xlab='tiempo(h)', ylab= 'Bioetanol (g/l)')
> par(new='TRUE')
> plot(x,f,type='l', xlim=c(s[1],s[n]), ylim=c(0,60), xlab='', ylab='', col='blue')
```



60
50
40
30
20
10
0
Bioetanol (g/l)
2
4
6
8
10
tiempo(h)