

Práctica 6

Objetivo: poner en práctica los conceptos adquiridos en las prácticas anteriores.

Ejercicio 1:

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 serán:

1. **Paso 1:** se deberá programar la función Polbase, para obtener las funciones de base la interpolación de Lagrange, que están dadas por la expresión:

$$L[i] = \prod_{j=1, j \neq i}^n \frac{t-s[j]}{s[i]-s[j]}$$

Para ello realizamos el siguiente algoritmo:

```
> Polbase=function(n,s,t){
+   L=0 # hay que ponerlo igual a 0 para hacerlo vector, aunque se use en un
+   productorio
+   for(i in 1:n){
+     L[i]=1 #ahora si se pone a 1
+
+     for(j in 1:n){
+
+       if (i!=j){ # !=i significa que j no tomará los valores de i
+
+
+         L[i]= L[i]*( t - s [j]) / (s[i]- s[j])
+       }
+     }
+   }
}
```

```

+ }
+
+ }
+ return(L) #Return es el resultado
+ }

```

2. **Paso 2:** programar la función PolInterp para obtener el polinomio interpolador en el punto t.

Los datos de entrada que necesitaremos serán:

- El vector B, que contendrá los valores de la función interpolada
- El vector L, que contendrá los polinomios de base evaluados en t
- La variable n

Para ello realizaremos el siguiente algoritmo:

```

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

```

3. **Paso 3;** introduciremos los datos, dados al principio, y ejecutaremos los algoritmos. Los valores obtenidos de las funciones, se almacenarán en el vector L (polinomio de base) y en la variable p (polinomio interpolador)

```

> #Datos
> B=c(10,20,35.33,35.5)
> s=c(1,3.5,5,10)
> n=length(s)
> #punto 1
> T1=2 #valor a interpolar, puede ser pepito, eliseo o lo que sea el nombre/ el nombre da igual, la función identifica 1º con 1º, 2º con 2º...IMPORTA EL ORDEN
> L1=Polbase(n,s,T1)
> L1
[1] 0.40000000 0.98461538 -0.40000000 0.01538462
> pol1=Polinterp(B,L1,n)
> pol1
[1] 10.10646
> #punto 2
> T2=6.5
> L2=Polbase(n,s,T2)
> L2
[1] 0.17500000 -1.18461538 1.92500000 0.08461538
> pol2=Polinterp(B,L2,n)
> pol2

```

[1] 49.07179

4. **Paso 4:** dibujaremos conjuntamente los valores obtenidos, para ello:

- Tendremos que obtener 1001 abscisas equidistantes en el intervalo $[s[1],s[n]]$ y almacenarlas en el vector x
- Llamaremos 1001 veces las funciones Polbase y PolInterp . PolInterp se almacenará en el vector $f[k]$, $k=1,\dots,1001$.
- Usaremos el comando $\text{"plot()}"$ para dibujar el gráfico

> **#Gráfica**

> **#para obtener 1001 abscisas equidistantes en $s[1]$, $s[n]$**

> $x=\text{seq}(s[1],s[n], \text{length.out} = 1001)$

> **#otra opcion es poner lo que mide cada trocito**

> $h = ((s[n]- s[1]) / 1000)$

> $x=\text{seq}(s[1],s[n],h)$

> **#y otra opción sería hacer un bucle:**

> for (i in 1:1001){

+ $x[i]=s[1]+(i-1)h$

> **#queremos obtener el valor del polinomio interpolador en puntos x , para poder dibujarlo**

> $f=0$

> for (k in 1:1001){

+ $T=x[k]$

+ $LL=\text{Polbase}(n,s,T)$

+ $f[k]=\text{Polinterp}(B,LL,n)$

+ }

>

> $\text{plot}(s,B,\text{xlim}=c(s[1],s[n]),\text{ylim}=c(0,60),\text{pch}=19,\text{col}='blue',$

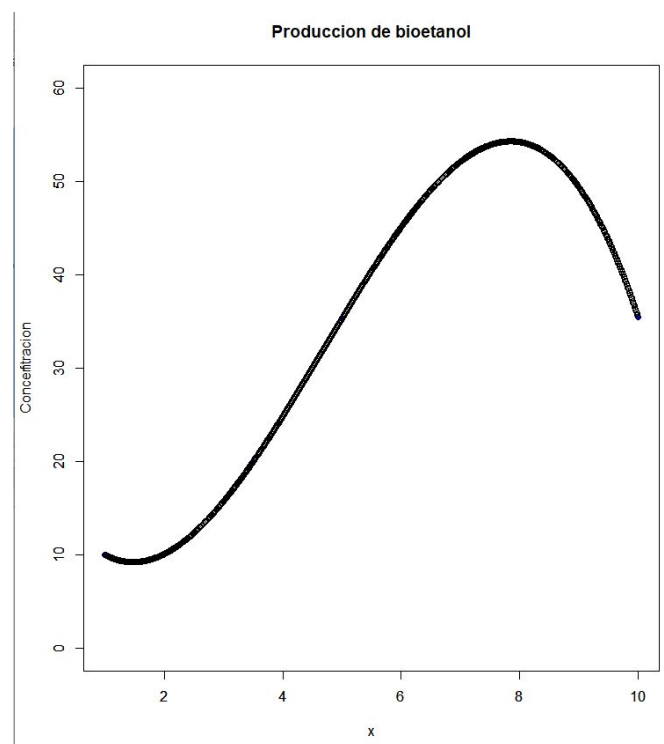
$\text{xlab}='x',\text{ylab}='Concentracion',\text{main}='Produccion de bioetanol')$

> $\text{par}(\text{new}='TRUE')$

> $\text{plot}(x,f, \text{xlim}=c(s[1],s[n]),\text{ylim}=c(0,60))$

#xlim e ylim son para poner la dos funciones en escala

Obtendremos el siguiente gráfico:



Errores:

- La función está definida por un sumatorio, asegúrate de inicializar el vector L, como $L=0$, FUERA del bucle for.
- Cada bucle que abres, ya sea condicional, while o for, DEBES CERRARLO. Las funciones TAMBIÉN.
- Ten cautela con los paréntesis, todos los que abres se deben cerrar.
- Si son vectores debes añadir el igual y la c como digo en la explicación. El vector tiene la siguiente: `vector=c(Datos)`
- Deberás poner `()` en el bucle para poner desde donde oscila n
- Se escribe `lenght` en lugar de `length`.
- Deberás poner el comando `return ()` tras la función
- `par=(new...)` es incorrecto, la forma correcta es `par(new...)`
- Deberás poner las llaves de los `if`, según se cierre el paréntesis
- Mantén el orden de las variables de las funciones descritas a la hora de realizar las operaciones. Ej: `Polbase=function(n,s,t)`

Video: <https://youtu.be/j-F5PyqshYY>