

EJERCICIO DE DERIVACIÓN

Calcula la primera derivada de la función $\text{sen}(x)$ en el punto 3.5 siendo el soporte de la función: 3, 6.5, 7, 4.7, 9 mediante el método de los coeficientes indeterminados.

Datos del problema:

- Función: $\text{sen}(x)$
- Soporte: 3, 6.5, 7, 4.7, 9.
- Punto de evaluación de la derivada: 3.5
- Orden de la derivada: 1

Pasos a seguir en R:

1. Utilizamos los comandos que limpian la *consola* y el *environment* para evitar fallos en R: `cat("\014") rm(list=ls())`
2. Introducimos las variables de entrada en R: estas variables serán los datos del problema.
 - Introduciremos el soporte como un vector con la función concatenar:
`s_points<-c(3, 6.5, 7, 4.7, 9)`
 - El punto de evaluación será: `x_0<-3.5`
 - El orden de la derivada es 1, por lo que: `orden_diff<-1`
 - La función es definida de la siguiente manera: `f<-function(x){ return(sin(x))}`
 - Calculamos el valor de la función en el soporte: `f_values<-f(s_points)`
3. Introducimos en R el programa que contiene la función `diff_indeterminados` para calcular los coeficientes de derivación haciendo uso de ella.
4. Calculamos los coeficientes de derivación y hacemos que se muestren en consola usando `print`.

```
coeficientes <-diff_indeterminados(s_points, x_0, orden_diff)
```

```
print(coeficientes)
```

5. Calculamos la derivada sabiendo que viene definida por la siguiente fórmula:

$$f^{(k)}(x^*) \approx \sum_{i=0}^n c_i \cdot f(s_i)$$

Para ello hacemos el sumatorio del producto de los coeficientes calculados anteriormente y la función evaluada en el soporte que también hemos calculado y guardado en la variable `f_values`. Utilizaremos la función `sum` presente en R de la siguiente manera:

```
derivada<-sum(coeficientes*f_values)
```

6. Mostramos en pantalla el resultado de la derivada calculada usando `cat`:

```
cat("La derivada de la función en x* es: ", derivada, "\n")
```

Programa en R:

```
1 cat("\014")
2 rm(list=ls())
3
4 s_points<-c(3, 6.5, 7, 4.7, 9)
5 x_0<-3.5
6 orden_diff<-1
7 f<-function(x){
8   return(sin(x))
9 }
10 f_values<-f(s_points)
11
12 diff_indeterminados <- function(s_points, x_0, orden_diff) {
13   abs_diff <- abs(s_points - x_0) #hacemos la resta del soporte y x* para calcular h
14   filtered_diff <- abs_diff[abs_diff != 0] #miramos que no haya ningún elemento en valor absoluto igual a 0
15
16   if (length(filtered_diff) == 0) {
17     stop()
18   } #nos aseguramos de que lo que nos queda no sea 0
19
20   h <- min(filtered_diff) #nos quedamos con la distancia más pequeña de todas
21   n_points = length(s_points)
22
23
24   # Validar que el orden de la derivada no exceda el número permitido
25   if (orden_diff >= n_points) {
26     stop(
27       paste(
28         "El orden de la derivada (",
29         orden_diff,
30         ") debe ser menor que el número de puntos (",
31         n_points,
32         ")."
33       )
34     )
35   }
36
37   thetas <- (s_points - x_0) / h #despejamos de la fórmula de la teoría
38   vandermonde_matrix <- matrix(0, nrow = n_points, ncol = n_points)
39   for (i in 1:n_points) {
40     for (j in 1:n_points) {
41       vandermonde_matrix[i, j] <- thetas[j] ^ (i - 1)
42     } #se eleva a la posición de la fila menos 1
43   }
44
45   # Crear el vector del lado derecho
46   diff_equal <- numeric(length = n_points)
47   diff_equal[orden_diff + 1] <- factorial(orden_diff)
48
49   coff_indet_diff <- solve(vandermonde_matrix, diff_equal * (1 / h ^ orden_diff))
50
51   return(coff_indet_diff)
52 }
53
54 coeficientes <- diff_indeterminados(s_points, x_0, orden_diff)
55 derivada<-sum(coeficientes*f_values)
56
57 cat("La derivada de la función en x* es: ", derivada, "\n")
```