

**UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO**  
**FACULTAD DE ESTADÍSTICA E INFORMÁTICA**  
**PROGRAMACIÓN NUMÉRICA**

**Diferenciación Numérica**

Serie de Ejercicios 8.1 al 8.7

**Estudiante:** Leydy Griselda Aguilar Ccopa

**Docente:** Fred Torres Cruz

**Curso:** Programación Numérica

**Ciclo:** 2025-II

Puno, Perú  
2025

# 1. Definición

La diferenciación numérica es una técnica usada para aproximar derivadas de una función cuando no se conoce su forma analítica o solo se tienen datos discretos (tablas o mediciones).

La derivada de una función  $f(x)$  mide qué tan rápido cambia la función en un punto:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Como en los datos reales no podemos hacer que  $h \rightarrow 0$ , se emplean **diferencias finitas**:

## 1.1. Diferencia hacia adelante

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Usa el punto actual y el siguiente. Ideal para el inicio de una tabla de datos.

## 1.2. Diferencia hacia atrás

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

Usa el punto actual y el anterior. Ideal para el final de una tabla.

## 1.3. Diferencia centrada

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Usa valores a ambos lados del punto, por lo que es más precisa.

## 1.4. Segunda derivada

La segunda derivada, que mide la aceleración o curvatura, se calcula como:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

## Ejercicio 8.1: Análisis de Crecimiento de Usuarios

Datos:

Mes	1	2	3	4	5	6	7
Usuarios (miles)	10	15	23	34	48	65	85

1) Tasa de crecimiento en el mes 4 (diferencia centrada):

$$f'(4) \approx \frac{f(5) - f(3)}{2h} = \frac{48 - 23}{2(1)} = 12,5$$

**Resultado:** 12,5 miles de usuarios por mes.

2) Tasa de crecimiento en el mes 1 (diferencia hacia adelante):

$$f'(1) \approx \frac{f(2) - f(1)}{h} = 15 - 10 = 5$$

**Resultado:** 5 miles de usuarios por mes.

3) Tasa de crecimiento en el mes 7 (diferencia hacia atrás):

$$f'(7) \approx \frac{f(7) - f(6)}{h} = 85 - 65 = 20$$

**Resultado:** 20 miles de usuarios por mes.

4) Aceleración del crecimiento (segunda derivada centrada):

$$f''(i) \approx \frac{f(i+1) - 2f(i) + f(i-1))}{h^2}$$

Mes	2	3	4	5	6
$f''(x)$	3	3	3	3	3

**Resultado:** La segunda derivada es constante e igual a 3 miles/mes<sup>2</sup>.

**5) Interpretación:** La tasa de crecimiento ( $f'$ ) aumenta con el tiempo, y la aceleración ( $f''$ ) es positiva y constante. Por lo tanto, la startup está creciendo de forma **acelerada**, con una **aceleración constante**.

Código en R:

Listing 1: Ejercicio 8.1: Análisis de Crecimiento de Usuarios

```
1 #
2 #
3 #
```

```
=====
# EJERCICIO 8.1: ANÁLISIS DE CRECIMIENTO DE USUARIOS
#
=====
```

```

4
5 cat("===== EJERCICIO 8.1 =====\n\n")
6
7 # Datos
8 meses <- 1:7
9 usuarios <- c(10, 15, 23, 34, 48, 65, 85) # en miles
10 h <- 1 # espaciado entre meses
11
12 # Crear dataframe
13 datos_usuarios <- data.frame(Mes = meses, Usuarios = usuarios)
14 print(datos_usuarios)
15 cat("\n")
16
17 # --- 1) Tasa de crecimiento en el mes 4 (diferencia centrada)
18 # ---
19 cat("1) Tasa de crecimiento en el mes 4 (diferencia centrada):\n"
20 )
21 i <- 4
22 tasa_mes4 <- (usuarios[i+1] - usuarios[i-1]) / (2*h)
23 cat(sprintf("f'(4) = (f(5) - f(3)) / (2h) = (%d - %d) / %d = %.2f\n",
24 usuarios[i+1], usuarios[i-1], 2*h, tasa_mes4))
25 cat(sprintf("Resultado: %.1f miles de usuarios por mes\n\n", tasa
26 _mes4))
27
28 # --- 2) Tasa de crecimiento en el mes 1 (diferencia hacia
29 adelante) ---
30 cat("2) Tasa de crecimiento en el mes 1 (diferencia hacia
31 adelante):\n")
32 tasa_mes1 <- (usuarios[2] - usuarios[1]) / h
33 cat(sprintf("f'(1) = (f(2) - f(1)) / h = (%d - %d) / %d = %.2f\n",
34 usuarios[2], usuarios[1], h, tasa_mes1))
35 cat(sprintf("Resultado: %.0f miles de usuarios por mes\n\n", tasa
36 _mes1))
37
38 # --- 3) Tasa de crecimiento en el mes 7 (diferencia hacia atr s
39 ) ---
40 cat("3) Tasa de crecimiento en el mes 7 (diferencia hacia atr s
41 :)\n")
42 tasa_mes7 <- (usuarios[7] - usuarios[6]) / h
43 cat(sprintf("f'(7) = (f(7) - f(6)) / h = (%d - %d) / %d = %.2f\n",
44 usuarios[7], usuarios[6], h, tasa_mes7))
45 cat(sprintf("Resultado: %.0f miles de usuarios por mes\n\n", tasa
46 _mes7))
47
48 # --- 4) Aceleraci n del crecimiento (segunda derivada centrada)
49 # ---
50 cat("4) Aceleraci n del crecimiento (segunda derivada centrada)
51 :)\n")

```

```

41 cat("f''(i) = (f(i+1) - 2f(i) + f(i-1)) / h \n\n")
42
43 meses_aceleracion <- 2:6
44 aceleracion <- numeric(length(meses_aceleracion))
45 for(j in 1:length(meses_aceleracion)) {
46   i <- meses_aceleracion[j]
47   aceleracion[j] <- (usuarios[i+1] - 2*usuarios[i] + usuarios[i
48     -1]) / h^2
49 }
50 resultados_aceleracion <- data.frame(Mes = meses_aceleracion,
51                                       Segunda_Derivada =
52                                         aceleracion)
53 print(resultados_aceleracion)
54 cat(sprintf("\nLa segunda derivada es constante: %.0f miles/mes
55   \n\n",
56           mean(aceleracion)))
57
58 # --- 5) Interpretaci n ---
59 cat("5) Interpretaci n:\n")
60 cat("- La tasa de crecimiento (primera derivada) aumenta con el
61   tiempo\n")
62 cat("- La aceleraci n (segunda derivada) es positiva y constante
63   ( 3 )\n")
64 cat("- Conclusi n: La startup est  creciendo de forma ACELERADA
65   ,\n")
66 cat("  con una ACELERACI N CONSTANTE\n\n")
67
68 # Gr fico
69 par(mfrow = c(1, 2), mar = c(4, 4, 3, 2))
70 plot(meses, usuarios, type = "b", pch = 19, col = "blue",
71      xlab = "Mes", ylab = "Usuarios (miles)",
72      main = "Crecimiento de Usuarios", lwd = 2)
73 grid()
74
75 plot(meses_aceleracion, aceleracion, type = "b", pch = 19, col =
76   "red",
77      xlab = "Mes", ylab = "f''(x) (miles/mes )",
78      main = "Aceleraci n del Crecimiento", lwd = 2, ylim = c(0,
79      5))
80 abline(h = 3, col = "red", lty = 2)
81 grid()
82 par(mfrow = c(1, 1))

```

## Ejercicio 8.2: Optimización de Función de Pérdida

Datos:

Época	0	10	20	30	40	50
Loss	2.45	1.82	1.35	1.08	0.95	0.89

1) Tasa de cambio del loss en la época 20:

$$L'(20) \approx \frac{L(30) - L(10)}{2h} = \frac{1,08 - 1,82}{20} = -0,037$$

Resultado:  $-0,037$  unidades de loss por época.

2) Segunda derivada en la época 30:

$$L''(30) = \frac{0,95 - 2(1,08) + 1,35}{100} = 0,0014$$

Resultado:  $L''(30) = 0,0014 > 0$ , indicando que la disminución del loss se desacelera (convergencia).

3) Época donde  $\Delta L < 0,01$ :

A partir de los cálculos, el criterio se cumple desde la **época 40**.

4) Estimación del loss en la época 25:

$$L(25) \approx L(20) + L'(20)(25 - 20) = 1,35 - 0,037(5) = 1,165$$

Resultado:  $L(25) \approx 1,165$

Código en R:

Listing 2: Ejercicio 8.2: Optimización de la Función de Pérdida

```
1 #  
   =====  
2 # EJERCICIO 8.2: OPTIMIZACION DE FUNCION DE PERDIDA  
3 #  
   =====  
4  
5 cat("\n\n===== EJERCICIO 8.2 =====\n\n")  
6  
7 epocas <- c(0, 10, 20, 30, 40, 50)  
8 loss <- c(2.45, 1.82, 1.35, 1.08, 0.95, 0.89)  
9 h <- 10  
10  
11 datos_loss <- data.frame(Epoca = epocas, Loss = loss)  
12 print(datos_loss)  
13 cat("\n")
```

```

14
15 # --- 1) Tasa de cambio del loss en la poca 20 ---
16 cat("1) Tasa de cambio del loss en la poca 20 (diferencia
centrada):\n")
17 i <- 3
18 tasa_loss_20 <- (loss[i+1] - loss[i-1]) / (2*h)
19 cat(sprintf("L'(20) = (L(30) - L(10)) / (2h) = (%.2f - %.2f) / %d
= %.4f\n",
20 loss[i+1], loss[i-1], 2*h, tasa_loss_20))
21 cat(sprintf("Resultado: %.3f unidades de loss por poca \n\n",
tasa_loss_20))
22
23 # --- 2) Segunda derivada en la poca 30 ---
24 cat("2) Segunda derivada en la poca 30:\n")
25 i <- 4
26 segunda_deriv_30 <- (loss[i+1] - 2*loss[i] + loss[i-1]) / h^2
27 cat(sprintf("L''(30) = (%.2f - 2(% .2f) + %.2f) / %d = %.6f\n",
28 loss[i+1], loss[i], loss[i-1], h^2, segunda_deriv_30)
)
29 cat(sprintf("\nResultado: L''(30) = %.4f > 0\n", segunda_deriv_
30))
30 cat("Interpretaci n: el modelo se aproxima a la convergencia.\n\
n")
31
32 # --- 3) poca donde Loss < 0.01 ---
33 cat("3) poca donde Loss < 0.01:\n\n")
34 n <- length(epocas)
35 intervalos <- data.frame(
36 Intervalo = character(n-1),
37 Delta_Loss_por_epoca = numeric(n-1),
38 Magnitud = numeric(n-1)
39 )
40 for(i in 1:(n-1)) {
41 delta_loss_por_epoca <- (loss[i+1] - loss[i]) / h
42 intervalos$Intervalo[i] <- sprintf("%d-%d", epocas[i], epocas[i
+1])
43 intervalos$Delta_Loss_por_epoca[i] <- delta_loss_por_epoca
44 intervalos$Magnitud[i] <- abs(delta_loss_por_epoca)
45 }
46 print(intervalos)
47 cat("\n")
48
49 epoca_parada <- which(intervalos$Magnitud < 0.01)[1]
50 if(!is.na(epoca_parada)) {
51 cat(sprintf("El criterio se cumple en la POCA %d\n",
52 epocas[epoca_parada + 1]))
53 }
54
55 # --- 4) Estimaci n del loss en la poca 25 ---
56 cat("\n4) Estimaci n del loss en la poca 25:\n")
57 epoca_target <- 25

```

```

58 loss_25 <- loss[3] + tasa_loss_20 * (epoca_target - epocas[3])
59 cat(sprintf("L(25) = %.2f + (%.4f)*5 = %.4f\n",
60           loss[3], tasa_loss_20, loss_25))
61 cat(sprintf("Resultado: L(25)      %.3f\n", loss_25))
62
63 par(mfrow = c(1, 2))
64 plot(epocas, loss, type = "b", col = "darkgreen", pch = 19,
65      main = "Funci n de P rdida", xlab = " poca ", ylab = "Loss
66      ", lwd = 2)
67 points(25, loss_25, pch = 4, col = "red", cex = 2, lwd = 2)
68 grid()
69 plot(epocas[1:(n-1)] + h/2, intervalos$Magnitud, type = "b", pch
70      = 19,
71      col = "purple", main = "Magnitud del Cambio del Loss",
72      xlab = " poca ", ylab = "| Loss / poca |", lwd = 2)
73 abline(h = 0.01, col = "red", lty = 2)
74 grid()
75 par(mfrow = c(1, 1))
76 cat("\n===== FIN =====\n")

```

## Ejercicio 8.3: Análisis de Series Temporales de Ventas

Datos:

Día	1 (Lun)	2 (Mar)	3 (Mié)	4 (Jue)	5 (Vie)	6 (Sáb)	7 (Dom)
Ventas (\$k)	45	52	61	58	73	89	95

1) Velocidad de crecimiento (derivada) para cada día:

- $f'(1) \approx (52 - 45)/1 = 7,0$  (Adelante)
- $f'(2) \approx (61 - 45)/2 = 8,0$  (Centrada)
- $f'(3) \approx (58 - 52)/2 = 3,0$  (Centrada)
- $f'(4) \approx (73 - 61)/2 = 6,0$  (Centrada)
- $f'(5) \approx (89 - 58)/2 = 15,5$  (Centrada)
- $f'(6) \approx (95 - 73)/2 = 11,0$  (Centrada)
- $f'(7) \approx (95 - 89)/1 = 6,0$  (Atrás)

2) Día con mayor aceleración (segunda derivada):

- $f''(2) = (61 - 2 * 52 + 45)/1^2 = 2$
- $f''(3) = (58 - 2 * 61 + 52)/1^2 = -12$
- $f''(4) = (73 - 2 * 58 + 61)/1^2 = 18$
- $f''(5) = (89 - 2 * 73 + 58)/1^2 = 1$
- $f''(6) = (95 - 2 * 89 + 73)/1^2 = -10$

**Resultado:** El Día 4 (Jueves) tuvo la mayor aceleración (18).

3) **Magnitud de la desaceleración del Jueves:** La caída en la tendencia se ve el Jueves, pero la desaceleración (fuerte  $f''$  negativa) se calcula en el Miércoles (Día 3).

$$f''(3) = \frac{58 - 2(61) + 52}{1} = -12$$

**Resultado:** La magnitud de la desaceleración fue de  $-12$  \$k/día<sup>2</sup>.

4) **Extrapolación para el Lunes siguiente (Día 8):** Usamos la última tasa de crecimiento (Día 7):  $f(8) \approx f(7) + f'(7) \times h$

$$f(8) \approx 95 + 6 \times 1 = 101$$

**Resultado:** Se esperarían \$101k en ventas.

**Código en R:**

Listing 3: Ejercicio 8.3: Análisis de Series Temporales de Ventas

```

1  #
   =====
2  # EJERCICIO 8.3: ANÁLISIS DE SERIES TEMPORALES DE VENTAS
3  #
   =====
4
5  cat("\n\n===== EJERCICIO 8.3 =====\n\n")
6
7  dias <- 1:7
8  ventas <- c(45, 52, 61, 58, 73, 89, 95)
9  h <- 1
10 n <- length(ventas)
11 nombres_dias <- c("Lun", "Mar", "Mie", "Jue", "Vie", "Sab", "Dom"
   )
12
13 datos_ventas <- data.frame(Dia = dias, Nombre = nombres_dias,
   Ventas = ventas)
14 print(datos_ventas)
15 cat("\n")
16
17 # --- 1) Velocidad de crecimiento (derivada) para cada día ---
18 cat("1) Velocidad (f') de crecimiento de ventas:\n")
19 f1_adelante <- (ventas[2] - ventas[1]) / h
20 f1_central <- (ventas[3:n] - ventas[1:(n-2)]) / (2*h)
21 f1_atras <- (ventas[n] - ventas[n-1]) / h
22 velocidad <- c(f1_adelante, f1_central, f1_atras)
23
24 df_velocidad <- data.frame(Dia = nombres_dias, Velocidad_Ventas_k
   = velocidad)
25 print(df_velocidad)
26 cat("\n")
27
28 # --- 2) Día con mayor aceleración (segunda derivada) ---
29 cat("2) Aceleración (f'') de ventas:\n")
30 aceleracion <- numeric(n-2)
31 for (i in 2:(n-1)) {
32     aceleracion[i-1] <- (ventas[i+1] - 2*ventas[i] + ventas[i-1])
       / h^2
33 }
34 df_aceleracion <- data.frame(Dia = nombres_dias[2:(n-1)],
   Aceleracion = aceleracion)
35 print(df_aceleracion)
36
37 max_acel_dia <- nombres_dias[which.max(aceleracion) + 1]
38 cat(sprintf("\nDía con mayor aceleración: %s (f'' = %.0f)\n\n",
   max_acel_dia, max(aceleracion)))
39
40
41 # --- 3) Magnitud de la desaceleración (Miércoles, víspera del

```

```

Jueves) ---
42 cat("3) Magnitud de la desaceleracion (calculada en Mie, Dia 3):\n")
43 f_pp_3 <- aceleracion[2] # f''(3)
44 cat(sprintf("La magnitud de la desaceleracion fue: %.0f $k/dia^2\n", f_pp_3))
45
46 # --- 4) Extrapolacion para el Lunes siguiente (Dia 8) ---
47 cat("4) Extrapolacion para Lunes (Dia 8):\n")
48 f_p_7 <- velocidad[n] # f'(7)
49 f_8 <- ventas[n] + f_p_7 * h
50 cat(sprintf("f(8) = f(7) + f'(7)*h = %.0f + %.0f * %d = %.0f\n",
51             ventas[n], f_p_7, h, f_8))
52 cat(sprintf("Resultado: Se esperan %.0f $k en ventas.\n\n", f_8))

```

## Ejercicio 8.4: Gradiente de Función de Activación

Datos:

$x$	-3.0	-2.0	-1.0	0.0	1.0	2.0	3.0
$\sigma(x)$	0.0474	0.1192	0.2689	0.5000	0.7311	0.8808	0.9526

1)  $\sigma'(0)$  usando diferencia centrada ( $h=1$ ):

$$\sigma'(0) \approx \frac{\sigma(1) - \sigma(-1)}{2h} = \frac{0,7311 - 0,2689}{2(1)} = 0,2311$$

2)  $\sigma'(-2)$  y  $\sigma'(2)$  usando diferencias centradas ( $h=1$ ):

$$\sigma'(-2) \approx \frac{\sigma(-1) - \sigma(-3)}{2h} = \frac{0,2689 - 0,0474}{2(1)} = 0,11075$$

$$\sigma'(2) \approx \frac{\sigma(3) - \sigma(1)}{2h} = \frac{0,9526 - 0,7311}{2(1)} = 0,11075$$

3) Comparación con derivada analítica  $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ :

- **x=0:** Analítica:  $0,5(1 - 0,5) = 0,25$ . Numérica: 0,2311.
- **x=-2:** Analítica:  $0,1192(1 - 0,1192) \approx 0,10499$ . Numérica: 0,11075.
- **x=2:** Analítica:  $0,8808(1 - 0,8808) \approx 0,10499$ . Numérica: 0,11075.

**Resultado:** Los valores numéricos son cercanos, pero  $h = 1$  es un paso muy grande.

4) **Tamaño de  $h$  recomendado:** Un  $h = 1$  es demasiado grande y produce errores notables. Para mayor precisión, se debería usar un  $h$  mucho más pequeño, como  $h = 0,01$  o  $h = 0,001$ .

5) **Simetría de la derivada:** La derivada es simétrica ( $f'(-x) = f'(x)$ ) porque la función sigmoide  $\sigma(x)$  es simétrica rotacionalmente alrededor del punto  $(0, 0,5)$ . La pendiente en  $x = -2$  es la misma que la pendiente en  $x = 2$ .

Código en R:

Listing 4: Ejercicio 8.4: Gradiente de Función de Activación

```
1 #
  =====
2 # EJERCICIO 8.4: GRADIENTE DE FUNCION DE ACTIVACION
3 #
  =====
4
5 cat("\n\n===== EJERCICIO 8.4 =====\n\n")
```

```

6
7 x <- c(-3.0, -2.0, -1.0, 0.0, 1.0, 2.0, 3.0)
8 sigma_x <- c(0.0474, 0.1192, 0.2689, 0.5000, 0.7311, 0.8808,
9             0.9526)
10 h <- 1 # El paso entre valores de x es 1
11 n <- length(x)
12
13 datos_sigma <- data.frame(x = x, sigma_x = sigma_x)
14 print(datos_sigma)
15 cat("\n")
16 # --- 1) sigma'(0) usando diferencia centrada ---
17 cat("1) Gradiente en x=0 (centrada):\n")
18 i_cero <- which(x == 0)
19 grad_0_num <- (sigma_x[i_cero + 1] - sigma_x[i_cero - 1]) / (2*h)
20 cat(sprintf("sigma'(0) num = (s(1) - s(-1)) / 2 = (%.4f - %.4f) /
21             2 = %.5f\n\n",
22             sigma_x[i_cero + 1], sigma_x[i_cero - 1], grad_0_num)
23 )
24 # --- 2) sigma'(-2) y sigma'(2) (centrada) ---
25 cat("2) Gradientes en x=-2 y x=2 (centrada):\n")
26 i_neg2 <- which(x == -2)
27 grad_neg2_num <- (sigma_x[i_neg2 + 1] - sigma_x[i_neg2 - 1]) / (2
28 *h)
29 cat(sprintf("sigma'(-2) num = (s(-1) - s(-3)) / 2 = (%.4f - %.4f) /
30             / 2 = %.5f\n\n",
31             sigma_x[i_neg2 + 1], sigma_x[i_neg2 - 1], grad_neg2_
32 num))
33 i_pos2 <- which(x == 2)
34 grad_pos2_num <- (sigma_x[i_pos2 + 1] - sigma_x[i_pos2 - 1]) / (2
35 *h)
36 cat(sprintf("sigma'(2) num = (s(3) - s(1)) / 2 = (%.4f - %.4f) /
37             2 = %.5f\n\n",
38             sigma_x[i_pos2 + 1], sigma_x[i_pos2 - 1], grad_pos2_
39 num))
40 # --- 3) Comparaci3n con derivada anal tica ---
41 cat("3) Comparacion con derivada analitica s'(x) = s(x)*(1-s(x))
42 :\n")
43 grad_analitico <- sigma_x * (1 - sigma_x)
44 df_comp <- data.frame(x = x,
45                       Numerico = c(NA, grad_neg2_num, NA, grad_0_
46 num, NA, grad_pos2_num, NA),
47                       Analitico = grad_analitico)
48 print(df_comp, digits=5)
49 cat("\nLos valores numericos son cercanos, pero h=1 es muy grande
50 .\n\n")
51 # --- 4) Tama o de h recomendado ---

```

```

45 cat("4) Tamano de h recomendado:\n")
46 cat("h=1 es demasiado grande. Se recomienda un h mucho menor (ej.
    0.01) para mayor precision.\n\n")
47
48 # --- 5) Simetria de la derivada ---
49 cat("5) Simetria de la derivada:\n")
50 cat("La derivada es simetrica ( $f'(-x) = f'(x)$ ), como se ve en f
    '(-2) y f'(2).\n")
51 cat("Esto ocurre porque la funcion sigmoide es simetrica
    rotacionalmente alrededor de (0, 0.5).\n\n")

```

## Ejercicio 8.5: Detección de Anomalías en Métricas de Sistema

Datos:

Hora	0	1	2	3	4	5	6	7
Latencia (ms)	120	125	128	135	280	290	275	155

1) Tasa de cambio (primera derivada) para cada hora:

- $f'(0) \approx (125 - 120)/1 = 5,0$  (Adelante)
- $f'(1) \approx (128 - 120)/2 = 4,0$  (Centrada)
- $f'(2) \approx (135 - 125)/2 = 5,0$  (Centrada)
- $f'(3) \approx (280 - 128)/2 = 76,0$  (Anomalía)
- $f'(4) \approx (290 - 135)/2 = 77,5$  (Anomalía)
- $f'(5) \approx (275 - 280)/2 = -2,5$  (Inicio recuperación)
- $f'(6) \approx (155 - 290)/2 = -67,5$  (Recuperación)
- $f'(7) \approx (155 - 275)/1 = -120,0$  (Atrás)

2) Pico de anomalía (cambio de signo en  $f''$ ):

- $f''(2) = (135 - 2 * 128 + 125)/1^2 = 4$
- $f''(3) = (280 - 2 * 135 + 128)/1^2 = 138$  (Inicio del salto)
- $f''(4) = (290 - 2 * 280 + 135)/1^2 = -135$  (Cambio de signo: PICO)
- $f''(5) = (275 - 2 * 290 + 280)/1^2 = -25$

**Resultado:** El pico (cambio de  $f''$  de + a -) ocurre en la **Hora 4**.

3) **Magnitud del salto brusco (Horas 3-4):** Usando diferencia hacia adelante en la Hora 3:  $f'(3) \approx (f(4) - f(3))/1$

$$f'(3) \approx (280 - 135)/1 = 145 \text{ ms/hora}$$

4) **Tasa de recuperación (a partir Hora 6):** En la Hora 6, la tasa (centrada) es  $-67,5$  ms/hora. En la Hora 7, la tasa (atrás) es  $-120,0$  ms/hora.

5) **Momentos de anomalía ( $|f'| > 50$  ms/hora):** Revisando los valores de la Tarea 1, las anomalías se detectan en:

- **Hora 3** ( $f' = 76,0$ )
- **Hora 4** ( $f' = 77,5$ )

- Hora 6 ( $f' = -67,5$ )
- Hora 7 ( $f' = -120,0$ )

### Código en R:

Listing 5: Ejercicio 8.5: Detección de Anomalías

```

1 #
2 # =====
3 # EJERCICIO 8.5: DETECCION DE ANOMALIAS
4 # =====
5
6 cat("\n\n===== EJERCICIO 8.5 =====\n\n")
7
8 horas <- 0:7
9 latencia <- c(120, 125, 128, 135, 280, 290, 275, 155)
10 h <- 1
11 n <- length(latencia)
12
13 datos_latencia <- data.frame(Hora = horas, Latencia_ms = latencia)
14 print(datos_latencia)
15 cat("\n")
16
17 # --- 1) Tasa de cambio (primera derivada) ---
18 cat("1) Tasa de cambio (f') para cada hora:\n")
19 f1_adelante <- (latencia[2] - latencia[1]) / h
20 f1_central <- (latencia[3:n] - latencia[1:(n-2)]) / (2*h)
21 f1_atras <- (latencia[n] - latencia[n-1]) / h
22 velocidad <- c(f1_adelante, f1_central, f1_atras)
23
24 df_velocidad <- data.frame(Hora = horas, Tasa_ms_hora = velocidad)
25 print(df_velocidad)
26 cat("\n")
27
28 # --- 2) Pico de anomalia (cambio de signo f'') ---
29 cat("2) Aceleracion (f'') y Pico de Anomalia:\n")
30 aceleracion <- numeric(n-2)
31 for (i in 2:(n-1)) {
32     aceleracion[i-1] <- (latencia[i+1] - 2*latencia[i] + latencia[i-1]) / h^2
33 }
34 df_aceleracion <- data.frame(Hora = horas[2:(n-1)], Aceleracion = aceleracion)
35 print(df_aceleracion)

```

```

36 # Encontrar el cambio de signo de + a -
37 pico_hora <- 0
38 for(i in 1:(length(acceleracion)-1)) {
39     if(acceleracion[i] > 0 && aceleracion[i+1] < 0) {
40         pico_hora <- horas[i+2]
41     }
42 }
43 cat(sprintf("\nEl pico (cambio de f'' de + a -) ocurre en la HORA
    %d\n\n", pico_hora))
44
45 # --- 3) Magnitud del salto (Hora 3-4) ---
46 cat("3) Magnitud del salto (f' adelante en Hora 3):\n")
47 salto_3_4 <- (latencia[5] - latencia[4]) / h
48 cat(sprintf("f'(3) [adelante] = (280 - 135) / 1 = %.0f ms/hora\n\
    n", salto_3_4))
49
50 # --- 4) Tasa de recuperacion (a partir Hora 6) ---
51 cat("4) Tasa de recuperacion (Hora 6 y 7):\n")
52 cat(sprintf("f'(6) [centrada] = %.1f ms/hora\n", velocidad[7]))
53 cat(sprintf("f'(7) [atras] = %.1f ms/hora\n\n", velocidad[8]))
54
55 # --- 5) Momentos de anomalia (|f'| > 50) ---
56 cat("5) Momentos de anomalia (|f'| > 50 ms/hora):\n")
57 umbral_anomalia <- 50
58 horas_anomalia <- horas[abs(velocidad) > umbral_anomalia]
59 print(horas_anomalia)
60 cat("\n")

```

## Ejercicio 8.6: Análisis de Tasa de Conversión

Datos:

Gasto (\$k)	0	5	10	15	20	25
Conversión (%)	2.1	3.8	5.2	6.1	6.7	7.0

1) ROI marginal (derivada) en cada punto:  $h = 5$

- $f'(0) \approx (3,8 - 2,1)/5 = 0,34$  (Adelante)
- $f'(5) \approx (5,2 - 2,1)/(2 * 5) = 0,31$  (Centrada)
- $f'(10) \approx (6,1 - 3,8)/10 = 0,23$  (Centrada)
- $f'(15) \approx (6,7 - 5,2)/10 = 0,15$  (Centrada)
- $f'(20) \approx (7,0 - 6,1)/10 = 0,09$  (Centrada)
- $f'(25) \approx (7,0 - 6,7)/5 = 0,06$  (Atrás)

2) Rango de gasto con ROI marginal ¿0.2%: Resultado: En los rangos de gasto de \$0k, \$5k y \$10k.

3) Segunda derivada en \$15k (rendimientos decrecientes):

$$f''(15) \approx \frac{f(20) - 2f(15) + f(10)}{h^2} = \frac{6,7 - 2(6,1) + 5,2}{5^2}$$

$$f''(15) \approx \frac{11,9 - 12,2}{25} = -0,012$$

Resultado: El valor es negativo, confirmando rendimientos decrecientes.

4) ¿Recomendaría aumentar el gasto más allá de \$25k? No. El ROI marginal en \$25k es  $f'(25) \approx 0,06$ . Esto significa que por cada \$1000 adicionales, la conversión solo aumenta 0.06%. El costo de la inversión es mucho mayor que el beneficio.

Código en R:

Listing 6: Ejercicio 8.6: Análisis de Tasa de Conversión

```
1 #  
  =====  
2 # EJERCICIO 8.6: ANÁLISIS DE TASA DE CONVERSIÓN  
3 #  
  =====  
4  
5 cat("\n\n===== EJERCICIO 8.6 =====\n\n")  
6  
7 gasto <- c(0, 5, 10, 15, 20, 25)
```

```

8 conversion <- c(2.1, 3.8, 5.2, 6.1, 6.7, 7.0)
9 h <- 5
10 n <- length(gasto)
11
12 datos_conversion <- data.frame(Gasto_k = gasto, Conversion_pct =
    conversion)
13 print(datos_conversion)
14 cat("\n")
15
16 # --- 1) ROI marginal (derivada) en cada punto ---
17 cat("1) ROI marginal (f') en cada punto:\n")
18 f1_adelante <- (conversion[2] - conversion[1]) / h
19 f1_central <- (conversion[3:n] - conversion[1:(n-2)]) / (2*h)
20 f1_atras <- (conversion[n] - conversion[n-1]) / h
21 roi_marginal <- c(f1_adelante, f1_central, f1_atras)
22
23 df_roi <- data.frame(Gasto_k = gasto, ROI_Marginal = roi_marginal
    )
24 print(df_roi, digits=3)
25 cat("\n")
26
27 # --- 2) Rango de gasto con ROI marginal > 0.2 ---
28 cat("2) Rango de gasto con ROI marginal > 0.2:\n")
29 gasto_ideal <- gasto[roi_marginal > 0.2]
30 cat(sprintf("El ROI es > 0.2 en los niveles de gasto: %s $k\n\n",
31     paste(gasto_ideal, collapse=", ")))
32
33 # --- 3) Segunda derivada en $15k ---
34 cat("3) Segunda derivada (f'') en $15k:\n")
35 i_15k <- which(gasto == 15)
36 f_pp_15k <- (conversion[i_15k+1] - 2*conversion[i_15k] +
    conversion[i_15k-1]) / h^2
37 cat(sprintf("f''(15) = (%.1f - 2*%.1f + %.1f) / %d = %.4f\n",
38     conversion[i_15k+1], conversion[i_15k], conversion[i_
    15k-1], h^2, f_pp_15k))
39 cat("El valor negativo confirma rendimientos decrecientes.\n\n")
40
41 # --- 4) Recomendara aumentar el gasto > $25k? ---
42 cat("4) Recomendacion para gasto > $25k:\n")
43 cat(sprintf("No. El ROI marginal en $25k es %.2f (muy bajo).\n",
    roi_marginal[n]))
44 cat("El beneficio adicional es casi nulo y no justifica el costo
    .\n\n")

```

## Ejercicio 8.7: Feature Engineering con Derivadas

Datos:

Tiempo (s)	0	1	2	3	4	5	6	7
Temp (°C)	20.1	20.3	20.8	21.5	22.6	24.2	26.1	28.5

**1) Feature: Velocidad (Primera Derivada):** Se crea una columna **Velocidad** con los valores (adelante, centrada, atrás).

- $f'(0) \approx (20,3 - 20,1)/1 = 0,20$
- $f'(1) \approx (20,8 - 20,1)/2 = 0,35$
- ...
- $f'(7) \approx (28,5 - 26,1)/1 = 2,40$

**2) Feature: Aceleración (Segunda Derivada):** Se crea una columna **Aceleracion** con NA en los bordes.

- $f''(1) \approx (20,8 - 2 * 20,3 + 20,1)/1^2 = 0,3$
- $f''(2) \approx (21,5 - 2 * 20,8 + 20,3)/1^2 = 0,2$
- ...

**3) Detección de Alerta (Velocidad  $\geq 0.8$  °C/s):** Se detecta la alerta en los tiempos: 3, 4, 5, 6, 7.

**4) Normalización Min-Max:** Las features **Velocidad** y **Aceleracion** se re-escalan al rango  $[0, 1]$ .

**5) Utilidad de las Features:** Estas features son útiles porque capturan la **dinámica** y el **comportamiento** del sistema, no solo su estado actual.

- **Velocidad:** Detecta cambios bruscos (ej. una falla de refrigeración).
- **Aceleración:** Detecta inestabilidad (ej. si el proceso se vuelve errático), sirviendo como alerta temprana.

Código en R:

Listing 7: Ejercicio 8.7: Feature Engineering con Derivadas

```
1 #  
  =====  
2 # EJERCICIO 8.7: FEATURE ENGINEERING CON DERIVADAS  
3 #  
  =====
```

```

4 cat("\n\n===== EJERCICIO 8.7 =====\n\n")
5
6 # --- 0. Configuraci n Inicial ---
7 tiempo <- 0:7
8 temp <- c(20.1, 20.3, 20.8, 21.5, 22.6, 24.2, 26.1, 28.5)
9 df <- data.frame(Tiempo = tiempo, Temp = temp)
10 h <- 1
11 n <- nrow(df)
12
13 cat("--- DATOS INICIALES ---\n")
14 print(df)
15 cat("\n")
16
17 # --- Tarea 1: Feature Velocidad (1ra Derivada) ---
18 cat("--- TAREA 1 y 2: DataFrame con Derivadas ---\n")
19 vel_t0 <- (df$Temp[2] - df$Temp[1]) / h
20 vel_central <- (df$Temp[3:n] - df$Temp[1:(n - 2)]) / (2 * h)
21 vel_t7 <- (df$Temp[n] - df$Temp[n - 1]) / h
22 df$Velocidad <- c(vel_t0, vel_central, vel_t7)
23
24 # --- Tarea 2: Feature Aceleraci n (2da Derivada) ---
25 accel_bordes <- NA
26 f_x_plus_h <- df$Temp[3:n]
27 f_x <- df$Temp[2:(n - 1)]
28 f_x_minus_h <- df$Temp[1:(n - 2)]
29 accel_central <- (f_x_plus_h - 2 * f_x + f_x_minus_h) / (h^2)
30 df$Aceleracion <- c(accel_bordes, accel_central, accel_bordes)
31
32 print(df)
33 cat("\n")
34
35 # --- Tarea 3: Detecci n de Alerta (> 0.8 C /s) ---
36 cat("--- TAREA 3: Detecci n de Alerta ---\n")
37 umbral_alerta <- 0.8
38 df$Alerta <- ifelse(is.na(df$Velocidad), FALSE, df$Velocidad >
39   umbral_alerta)
40 momentos_alerta <- df$Tiempo[df$Alerta]
41
42 cat(paste("Momentos (segundos) con Velocidad >", umbral_alerta, "
43   :\n"))
44 print(momentos_alerta)
45 cat("\n")
46
47 # --- Tarea 4: Normalizaci n Min-Max ---
48 cat("--- TAREA 4: DataFrame Final con Normalizaci n ---\n")
49 min_max_scaler <- function(x) {
50   (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.
51     rm = TRUE))
52 }
53 df$Velocidad_Norm <- min_max_scaler(df$Velocidad)
54 df$Aceleracion_Norm <- min_max_scaler(df$Aceleracion)

```

```
52
53 print(df, digits=4)
54 cat("\n")
55
56 # --- Tarea 5: Explicaci n (impresa en consola) ---
57 cat("--- TAREA 5: Utilidad de las Features Derivadas ---\n")
58 cat("1. Velocidad: Detecta cambios bruscos (anomalias de
    comportamiento).\n")
59 cat("2. Aceleracion: Detecta inestabilidad (alerta temprana).\n")
60 cat("\n")
```