

PROGRAMACIÓN NUMÉRICA

Job Edward Apaza Curtihuanca

LOS EJERCICIOS SON DEL TRABAJO ENCARGADO EL 5/11/2025,
DONDE TENEMOS QUE RESOLVER EJERCICIOS, DEL PAPER QUE
NOS DIERON LA DISPOCICION

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/mes².

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**.

```

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| cat("1) Tasa de crecimiento en el mes 4 (diferencia centrada):\n")
19| i <- 4
20| tasa_mes4 <- (usuarios[i+1] - usuarios[i-1]) / (2*h)
21| cat(sprintf("f'(4) = (f(5) - f(3)) / (2h) = (%d - %d) / %d = %.2f\n",
22|           usuarios[i+1], usuarios[i-1], 2*h, tasa_mes4))
23| cat(sprintf("Resultado: %.1f miles de usuarios por mes\n\n", tasa_
mes4))
24|
25| # --- 2) Tasa de crecimiento en el mes 1 (diferencia hacia adelante)
26|     ---
27| cat("2) Tasa de crecimiento en el mes 1 (diferencia hacia adelante)
28|       :\n")
29| tasa_mes1 <- (usuarios[2] - usuarios[1]) / h
30| cat(sprintf("f'(1) = (f(2) - f(1)) / h = (%d - %d) / %d = %.2f\n",
31|           usuarios[2], usuarios[1], h, tasa_mes1))
32| cat(sprintf("Resultado: %.0f miles de usuarios por mes\n\n", tasa_
mes1))
33|
34| # --- 3) Tasa de crecimiento en el mes 7 (diferencia hacia atr s)
35|     ---
36| cat("3) Tasa de crecimiento en el mes 7 (diferencia hacia atr s):\n"
37|       ")
38| tasa_mes7 <- (usuarios[7] - usuarios[6]) / h
39| cat(sprintf("f'(7) = (f(7) - f(6)) / h = (%d - %d) / %d = %.2f\n",
40|           usuarios[7], usuarios[6], h, tasa_mes7))
41| cat(sprintf("Resultado: %.0f miles de usuarios por mes\n\n", tasa_
mes7))
42|
43| # --- 4) Aceleracion del crecimiento (segunda derivada centrada)
44|     ---
45| cat("4) Aceleracion del crecimiento (segunda derivada centrada):\n"
46|       )
47| cat("f''(i) = (f(i+1) - 2f(i) + f(i-1)) / h \n\n")
48|
49| meses_aceleracion <- 2:6
50| aceleracion <- numeric(length(meses_aceleracion))
51| for(j in 1:length(meses_aceleracion)) {

```

```

46 i <- meses_aceleracion[j]
47 aceleracion[j] <- (usuarios[i+1] - 2*usuarios[i] + usuarios[i-1])
48   / h^2
49 }
50
51 resultados_aceleracion <- data.frame(Mes = meses_aceleracion,
52                                         Segunda_Derivada = aceleracion)
53 print(resultados_aceleracion)
54 cat(sprintf("\nLa segunda derivada es constante: %.0f miles/mes \n",
55           mean(aceleracion)))
56
57 # --- 5) Interpretaci n ---
58 cat("5) Interpretaci n:\n")
59 cat("- La tasa de crecimiento (primera derivada) aumenta con el
60     tiempo\n")
61 cat("- La aceleraci n (segunda derivada) es positiva y constante (
62     3 )\n")
63 cat("- Conclusi n: La startup est    creciendo de forma ACELERADA,\n")
64 cat("    con una ACELERACI N CONSTANTE\n\n")
65
66 # Gr fico
67 par(mfrow = c(1, 2), mar = c(4, 4, 3, 2))
68 plot(meses, usuarios, type = "b", pch = 19, col = "blue",
69       xlab = "Mes", ylab = "Usuarios (miles)",
70       main = "Crecimiento de Usuarios", lwd = 2)
71 grid()
72
73 plot(meses_aceleracion, aceleracion, type = "b", pch = 19, col = "red",
74       xlab = "Mes", ylab = "f'(x) (miles/mes )",
75       main = "Aceleraci n del Crecimiento", lwd = 2, ylim = c(0, 5))
76 abline(h = 3, col = "red", lty = 2)
77 grid()
78 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 cat("\n\n===== EJERCICIO 8.2 =====\n\n")
5
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 = epochas, 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",
     loss[i+1], loss[i-1], 2*h, tasa_loss_20))
20 cat(sprintf("Resultado: %.3f unidades de loss por poca \\n\\n", tasa_
     loss_20))
21
22 # --- 2) Segunda derivada en la poca 30 ---
23 cat("2) Segunda derivada en la poca 30:\\n")
24 i <- 4
25 segunda_deriv_30 <- (loss[i+1] - 2*loss[i] + loss[i-1]) / h^2
26 cat(sprintf("L''(30) = (%.2f - 2%.2f) + %.2f) / %d = %.6f\\n",
     loss[i+1], loss[i], loss[i-1], h^2, segunda_deriv_30))
27 cat(sprintf("\\nResultado: L''(30) = %.4f > 0\\n", segunda_deriv_30))
28 cat("Interpretacion: el modelo se aproxima a la convergencia.\\n\\n")
29
30 # --- 3) poca donde Loss < 0.01 ---
31 cat("3) poca donde Loss < 0.01:\\n\\n")
32 n <- length(epochas)
33 intervalos <- data.frame(
34   Intervalo = character(n-1),
35   Delta_Loss_por_epoca = numeric(n-1),
36   Magnitud = numeric(n-1)
37 )
38
39 for(i in 1:(n-1)) {
40   delta_loss_por_epoca <- (loss[i+1] - loss[i]) / h
41   intervalos$Intervalo[i] <- sprintf("%d-%d", epochas[i], epochas[i
42     +1])
43   intervalos$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 POGA %d\\n",
52             epochas[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)
66 points(25, loss_25, pch = 4, col = "red", cex = 2, lwd = 2)
67 grid()
68
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)
72 abline(h = 0.01, col = "red", lty = 2)
73 grid()
74 par(mfrow = c(1, 1))
75 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)
- $f'(2) \approx (61 - 52)/2 = 8,0$ (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 $-12 \text{ } \$\text{k/día}^2$.

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  =====
3 # EJERCICIO 8.3: AN LISIS DE SERIES TEMPORALES DE VENTAS
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
14     = ventas)
15 print(datos_ventas)
16 cat("\n")
17
18 # --- 1) Velocidad de crecimiento (derivada) para cada d a ---
19 cat("1) Velocidad (f') de crecimiento de ventas:\n")
20 f1_adelante <- (ventas[2] - ventas[1]) / h
21 f1_central <- (ventas[3:n] - ventas[1:(n-2)]) / (2*h)
22 f1_atras <- (ventas[n] - ventas[n-1]) / h
23 velocidad <- c(f1_adelante, f1_central, f1_atras)
24
25 df_velocidad <- data.frame(Dia = nombres_dias, Velocidad_Ventas_k =
26     velocidad)
27 print(df_velocidad)
28 cat("\n")
29
30 # --- 2) D a con mayor aceleraci n (segunda derivada) ---
31 cat("2) Aceleracion (f'') de ventas:\n")
32 aceleracion <- numeric(n-2)
33 for (i in 2:(n-1)) {
34     aceleracion[i-1] <- (ventas[i+1] - 2*ventas[i] + ventas[i-1]) /
35         h^2
36 }
37 df_aceleracion <- data.frame(Dia = nombres_dias[2:(n-1)],
38     Aceleracion = aceleracion)
39 print(df_aceleracion)
40
41 max_acel_dia <- nombres_dias[which.max(aceleracion) + 1]
42 cat(sprintf("\nDia con mayor aceleracion: %s (f'' = %.0f)\n\n",

```

```

39         max_acel_dia, max(acceleracion)))
40
41 # --- 3) Magnitud de la desaceleracion (Miércoles, v_spera del
42     Jueves) ---
43 cat("3) Magnitud de la desaceleracion (calculada en Miércoles, Día 3):\n")
44 f_pp_3 <- acceleracion[2] # f'(3)
45 cat(sprintf("La magnitud de la desaceleracion fue: %.0f $k/dia^2\n\n",
46         " ", f_pp_3))
47
48 # --- 4) Extrapolación para el Lunes siguiente (Día 8) ---
49 cat("4) Extrapolación para Lunes (Día 8):\n")
50 f_p_7 <- velocidad[n] # f'(7)
51 f_8 <- ventas[n] + f_p_7 * h
52 cat(sprintf("f(8) = f(7) + f'(7)*h = %.0f + %.0f * %d = %.0f\n",
53             ventas[n], f_p_7, h, f_8))
54 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, 0.9526)
9  h <- 1 # El paso entre valores de x es 1
10 n <- length(x)
11
12 datos_sigma <- data.frame(x = x, sigma_x = sigma_x)
13 print(datos_sigma)
14 cat("\n")
15
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) / 2
21 = %.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*h)
28 cat(sprintf("sigma'(-2) num = (s(-1) - s(-3)) / 2 = (%.4f - %.4f) /
29 = %.5f\n",
29         sigma_x[i_neg2 + 1], sigma_x[i_neg2 - 1], grad_neg2_num))
30
31 i_pos2 <- which(x == 2)
32 grad_pos2_num <- (sigma_x[i_pos2 + 1] - sigma_x[i_pos2 - 1]) / (2*h)
33 cat(sprintf("sigma'(2) num = (s(3) - s(1)) / 2 = (%.4f - %.4f) / 2 =
34 %.5f\n\n",
34         sigma_x[i_pos2 + 1], sigma_x[i_pos2 - 1], grad_pos2_num))
35
36 # --- 3) Comparacion con derivada analitica ---
37 cat("3) Comparacion con derivada analitica s'(x) = s(x)*(1-s(x)):\n")
38 grad_analitico <- sigma_x * (1 - sigma_x)
39 df_comp <- data.frame(x = x,
40                         Numerico = c(NA, grad_neg2_num, NA, grad_0_num
41                                     , NA, grad_pos2_num, NA),
42                         Analitico = grad_analitico)
43 print(df_comp, digits=5)

```

```

42 cat("\nLos valores numericos son cercanos, pero h=1 es muy grande.\n
     \n")
43
44 # --- 4) Tamaño de h recomendado ---
45 cat("4) Tamaño 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 simétrica ( $f'(-x) = f'(x)$ ), como se ve en  $f'(-2)$ 
     y  $f'(2)$ .\n")
51 cat("Esto ocurre porque la función sigmoide es simétrica
     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)
- $f'(1) \approx (128 - 125)/2 = 4,0$ (Centrada)
- $f'(2) \approx (135 - 128)/2 = 5,0$ (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
4
5  cat("\n===== EJERCICIO 8.5 =====\n")
6
7  horas <- 0:7
8  latencia <- c(120, 125, 128, 135, 280, 290, 275, 155)
9  h <- 1
10 n <- length(latencia)
11
12 datos_latencia <- data.frame(Hora = horas, Latencia_ms = latencia)
13 print(datos_latencia)
14 cat("\n")
15
16 # --- 1) Tasa de cambio (primera derivada) ---
17 cat("1) Tasa de cambio (f') para cada hora:\n")
18 f1_adelante <- (latencia[2] - latencia[1]) / h
19 f1_central <- (latencia[3:n] - latencia[1:(n-2)]) / (2*h)
20 f1_atras <- (latencia[n] - latencia[n-1]) / h
21 velocidad <- c(f1_adelante, f1_central, f1_atras)
22
23 df_velocidad <- data.frame(Hora = horas, Tasa_ms_hora = velocidad)
24 print(df_velocidad)
25 cat("\n")
26
27 # --- 2) Pico de anomalia (cambio de signo f'') ---
28 cat("2) Aceleracion (f'') y Pico de Anomalia:\n")
29 aceleracion <- numeric(n-2)
30 for (i in 2:(n-1)) {
31   aceleracion[i-1] <- (latencia[i+1] - 2*latencia[i] + latencia[i-1]) / h^2

```

```

32 }
33 df_aceleracion <- data.frame(Hora = horas[2:(n-1)], Aceleracion =
34   aceleracion)
35
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 && acceleracion[i+1] < 0) {
40     pico_hora <- horas[i+2] # El indice i es para acceleracion,
41       que empieza en hora 2
42   }
43 cat(sprintf("\nEl pico (cambio de + a -) ocurre en la HORA %d
44 \n\n", pico_hora))
45
46 # --- 3) Magnitud del salto (Hora 3-4) ---
47 cat("3) Magnitud del salto (f' adelante en Hora 3):\n")
48 salto_3_4 <- (latencia[5] - latencia[4]) / h
49 cat(sprintf("f'(3) [adelante] = (280 - 135) / 1 = %.0f ms/hora\n\n",
50   salto_3_4))
51
52 # --- 4) Tasa de recuperacion (a partir Hora 6) ---
53 cat("4) Tasa de recuperacion (Hora 6 y 7):\n")
54 cat(sprintf("f'(6) [entrada] = %.1f ms/hora\n", velocidad[7]))
55 cat(sprintf("f'(7) [tras] = %.1f ms/hora\n\n", velocidad[8]))
56
57 # --- 5) Momentos de anomalia (|f'| > 50) ---
58 cat("5) Momentos de anomalia (|f'| > 50 ms/hora):\n")
59 umbral_anomalia <- 50
60 horas_anomalia <- horas[abs(velocidad) > umbral_anomalia]
61 print(horas_anomalia)
62 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 $\geq 0.2\%$: 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: AN LISIS DE TASA DE CONVERSI N  
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 # El paso es 5k
10 n <- length(gasto)
11
12 datos_conversion <- data.frame(Gasto_k = gasto, Conversion_pct =
13   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 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[
37   i_15k-1]) / h^2
38 sprintf("f''(15) = (%.1f - 2*%.1f + %.1f) / %d = %.4f\n",
39         conversion[i_15k+1], conversion[i_15k], conversion[i_15k-
40           1], h^2, f_pp_15k)
41 cat("El valor negativo confirma rendimientos decrecientes.\n\n")
42
43 # --- 4) Recomendara aumentar el gasto > $25k? ---
44 cat("4) Recomendacion para gasto > $25k:\n")
45 cat(sprintf("No. El ROI marginal en $25k es %.2f (muy bajo).\n", roi_
46   _marginal[n]))
47 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 > 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  =====
3
4 # EJERCICIO 8.7: FEATURE ENGINEERING CON DERIVADAS
5 #
6 =====
7
8 cat("\n\n===== EJERCICIO 8.7 =====\n\n")
9
10 # --- 0. Configuraci n Inicial ---
11 tiempo <- 0:7
12 temp <- c(20.1, 20.3, 20.8, 21.5, 22.6, 24.2, 26.1, 28.5)
13 df <- data.frame(Tiempo = tiempo, Temp = temp)
14 h <- 1
15 n <- nrow(df)
16
17
18 # --- Tarea 1: Feature Velocidad (1ra Derivada) ---
19 cat("--- TAREA 1 y 2: DataFrame con Derivadas ---\n")
20 vel_t0 <- (df$Temp[2] - df$Temp[1]) / h
21 vel_central <- (df$Temp[3:n] - df$Temp[1:(n - 2)]) / (2 * h)
22 vel_t7 <- (df$Temp[n] - df$Temp[n - 1]) / h
23 df$Velocidad <- c(vel_t0, vel_central, vel_t7)
24
25 # --- Tarea 2: Feature Aceleraci n (2da Derivada) ---
26 accel_bordes <- NA
27 f_x_plus_h <- df$Temp[3:n]
28 f_x <- df$Temp[2:(n - 1)]
29 f_x_minus_h <- df$Temp[1:(n - 2)]
30 accel_central <- (f_x_plus_h - 2 * f_x + f_x_minus_h) / (h^2)
31 df$Aceleracion <- c(accel_bordes, accel_central, accel_bordes)
32
33 print(df)
34 cat("\n")
35
36
37 # --- Tarea 3: Detecci n de Alerta (> 0.8 C /s) ---
38 cat("--- TAREA 3: Detecci n de Alerta ---\n")
39 umbral_alerta <- 0.8
40 df$Alerta <- ifelse(is.na(df$Velocidad), FALSE, df$Velocidad >
    umbral_alerta)
41 momentos_alerta <- df$Tiempo[df$Alerta]
```

```

42
43 cat(paste("Momentos (segundos) con Velocidad >", umbral_alerta, ":\n"
44   ""))
45 print(momentos_alerta)
46 cat("\n")
47
48 # --- Tarea 4: Normalizaci n Min-Max ---
49 cat("--- TAREA 4: DataFrame Final con Normalizaci n ---\n")
50 min_max_scaler <- function(x) {
51   (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm
52   = TRUE))
53 }
54 df$Velocidad_Norm <- min_max_scaler(df$Velocidad)
55 df$Aceleracion_Norm <- min_max_scaler(df$Aceleracion)
56
57 print(df, digits=4)
58 cat("\n")
59
60 # --- Tarea 5: Explicaci n (impresa en consola) ---
61 cat("--- TAREA 5: Utilidad de las Features Derivadas ---\n")
62 cat("1. Velocidad: Detecta cambios bruscos (anomalias de
63   comportamiento).\n")
64 cat("2. Aceleracion: Detecta inestabilidad (alerta temprana).\n")
65 cat("\n")

```