



# UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO

FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA  
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E  
INFORMÁTICA



## Diferenciación Numerica

**CURSO:** PROGRAMACIÓN NUMÉRICA

**DOCENTE:** FRED TORRES CRUZ

**PROPIO DE:**

- Cutipa Ramos, Nayelin Brisbane

**SEMESTRE:** Cuarto

**GRUPO:** "A"

**PUNO – PERÚ**

**2025 II**



# ÍNDICE

	Página
<b>1. Ejercicio 1: Análisis de Crecimiento de Usuarios</b>	<b>4</b>
1.1. Solución del Ejercicio 1 . . . . .	4
1.1.1. Datos Iniciales del Problema . . . . .	4
1.1.2. Tareas 1, 2 y 3: Tasa de Crecimiento (Primera Derivada) . . .	4
1.1.3. Tarea 4: Aceleración del Crecimiento (Segunda Derivada) . .	5
1.1.4. Tarea 5: Interpretación General . . . . .	6
1.2. Código en R para Verificación . . . . .	6
1.3. Salida de la Consola . . . . .	7
<b>2. Ejercicio 2: Optimización de Función de Pérdida</b>	<b>8</b>
2.1. Solución del Ejercicio 2 . . . . .	8
2.1.1. Tarea 1: Tasa de cambio en la época 20 (Diferencia Centrada)	9
2.1.2. Tarea 2: Segunda derivada en la época 30 . . . . .	9
2.1.3. Tarea 3: Criterio de parada ( $ f'  < 0,01$ ) . . . . .	9
2.1.4. Tarea 4: Estimación del loss en época 25 (Interpolación Lineal)	10
2.2. Código en R para Verificación . . . . .	10
2.3. Salida de la Consola . . . . .	12
<b>3. Ejercicio 3: Análisis de Series Temporales de Ventas</b>	<b>12</b>
3.1. Solución del Ejercicio 3 . . . . .	13
3.1.1. Tarea 1: Velocidad de Crecimiento (Primera Derivada) . . . .	13
3.1.2. Tarea 2: Día con Mayor Aceleración (Segunda Derivada) . . .	14
3.1.3. Tarea 3: Magnitud de la Caída el Jueves . . . . .	14
3.1.4. Tarea 4: Estimación Ventas Lunes Siguiete . . . . .	14
3.2. Código en R para Verificación . . . . .	14
3.3. Salida de la Consola . . . . .	16
<b>4. Ejercicio 4: Gradiente de Función de Activación</b>	<b>17</b>
4.1. Solución del Ejercicio 4 . . . . .	17
4.1.1. Tarea 1: Cálculo de $\sigma'(0)$ . . . . .	17
4.1.2. Tarea 2: Cálculo de $\sigma'(-2)$ y $\sigma'(2)$ . . . . .	18
4.1.3. Tarea 3: Comparación con la Derivada Analítica . . . . .	18
4.1.4. Tarea 4: Recomendación del tamaño de $h$ . . . . .	18
4.1.5. Tarea 5: Simetría de la Derivada . . . . .	19
4.2. Código en R para Verificación . . . . .	19
4.3. Salida de la Consola . . . . .	20
<b>5. Ejercicio 5: Detección de Anomalías en Métricas de Sistema</b>	<b>21</b>
5.1. Solución del Ejercicio 5 . . . . .	22
5.1.1. Tarea 1: Primera Derivada (Tasa de Cambio) . . . . .	22
5.1.2. Tarea 2: Pico de Anomalía (Segunda Derivada) . . . . .	22
5.1.3. Tarea 3: Magnitud del Salto (Horas 3 y 4) . . . . .	23
5.1.4. Tarea 4: Tasa de Recuperación (a partir Hora 6) . . . . .	23
5.1.5. Tarea 5: Detección de Anomalías (Umbral $> 50$ ms/h) . . . .	23
5.2. Código en R para Verificación . . . . .	23



5.3. Salida de la Consola . . . . .	25
<b>6. Ejercicio 6: Análisis de Tasa de Conversión</b>	<b>26</b>
6.1. Solución del Ejercicio 6 . . . . .	26
6.1.1. Tarea 1: ROI Marginal (Primera Derivada) . . . . .	26
6.1.2. Tarea 2: Rango de Gasto con ROI >0.2 %/\$k . . . . .	27
6.1.3. Tarea 3: Segunda Derivada en \$15k . . . . .	28
6.1.4. Tarea 4: Recomendación sobre Aumentar Gasto . . . . .	28
6.2. Código en R para Verificación . . . . .	28
6.3. Salida de la Consola . . . . .	30
<b>7. Ejercicio 7: Feature Engineering con Derivadas</b>	<b>30</b>
7.1. Solución del Ejercicio 7 . . . . .	31
7.1.1. Tarea 1: Feature 1 - Velocidad de Cambio (1ª Derivada) . . . . .	31
7.1.2. Tarea 2: Feature 2 - Aceleración (2ª Derivada) . . . . .	31
7.1.3. Tarea 3: Detección de Alertas ( $f'(t) > 0,8 \text{ }^\circ\text{C/s}$ ) . . . . .	32
7.1.4. Tarea 4: Normalización Min-Max . . . . .	32
7.1.5. Tarea 5: Utilidad para Modelos de Anomalías . . . . .	32
7.2. Código en R para Verificación . . . . .	32
7.3. Salida de la Consola . . . . .	34



## 1. Ejercicio 1: Análisis de Crecimiento de Usuarios

Una startup de tecnología registra el número acumulado de usuarios activos mensuales (en miles):

Cuadro 1: Usuarios acumulados por mes (en miles)

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

### Tareas:

1. Calcula la tasa de crecimiento de usuarios (usuarios nuevos por mes) en el mes 4 usando diferencia centrada.
2. Calcula la tasa de crecimiento en el mes 1 (usa diferencia adelante).
3. Calcula la tasa de crecimiento en el mes 7 (usa diferencia atrás).
4. ¿En qué mes fue mayor la aceleración del crecimiento? (calcula la segunda derivada en cada punto).
5. Interpreta: ¿la startup está creciendo de forma acelerada o desacelerada?

### 1.1. Solución del Ejercicio 1

#### 1.1.1. Datos Iniciales del Problema

Se definen los datos como una función  $f(x)$  donde  $x$  es el mes.

- $f(1) = 10$
- $f(2) = 15$
- $f(3) = 23$
- $f(4) = 34$
- $f(5) = 48$
- $f(6) = 65$
- $f(7) = 85$
- El paso (distancia entre puntos) es constante:  $h = 1$  mes.

#### 1.1.2. Tareas 1, 2 y 3: Tasa de Crecimiento (Primera Derivada)

##### 1. Tasa en Mes 4 (Diferencia Centrada)

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

- **Cálculo:**

$$f'(4) \approx \frac{f(5) - f(3)}{2 \cdot 1} = \frac{48 - 23}{2} = \frac{25}{2} = 12,5$$



- **Resultado: 12.5 mil usuarios / mes.**
- **Interpretación:** En el mes 4, la tasa de crecimiento era de 12,500 usuarios/mes.

## 2. Tasa en Mes 1 (Diferencia Hacia Adelante)

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

- **Cálculo:**

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

- **Resultado: 5 mil usuarios / mes.**
- **Interpretación:** Al inicio (mes 1), la tasa de crecimiento era de 5,000 usuarios/mes.

## 3. Tasa en Mes 7 (Diferencia Hacia Atrás)

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

- **Cálculo:**

$$f'(7) \approx \frac{f(7) - f(6)}{1} = 85 - 65 = 20$$

- **Resultado: 20 mil usuarios / mes.**
- **Interpretación:** Al final (mes 7), la tasa de crecimiento era de 20,000 usuarios/mes.

### 1.1.3. Tarea 4: Aceleración del Crecimiento (Segunda Derivada)

Se calcula la segunda derivada (aceleración) en cada punto usando la fórmula de 3 puntos.

- **Fórmula (con  $h = 1$ ):**  $f''(x) \approx f(x+1) - 2f(x) + f(x-1)$
- **Cálculos:** Se usa la fórmula centrada para puntos interiores y las fórmulas unilaterales de 3 puntos (que resultan equivalentes) para los extremos.

$$f''(1) \approx f(3) - 2f(2) + f(1) = 23 - 2(15) + 10 = 3$$

$$f''(2) \approx f(3) - 2f(2) + f(1) = 23 - 2(15) + 10 = 3$$

$$f''(3) \approx f(4) - 2f(3) + f(2) = 34 - 2(23) + 15 = 3$$

$$f''(4) \approx f(5) - 2f(4) + f(3) = 48 - 2(34) + 23 = 3$$

$$f''(5) \approx f(6) - 2f(5) + f(4) = 65 - 2(48) + 34 = 3$$

$$f''(6) \approx f(7) - 2f(6) + f(5) = 85 - 2(65) + 48 = 3$$

$$f''(7) \approx f(7) - 2f(6) + f(5) = 85 - 2(65) + 48 = 3$$

- **Resultado:** La aceleración del crecimiento fue **constante e igual en todos los meses**, con un valor de **3 mil usuarios / mes<sup>2</sup>**.



#### 1.1.4. Tarea 5: Interpretación General

**¿La startup está creciendo de forma acelerada o desacelerada?**

Dado que la segunda derivada (la aceleración) es **positiva y constante** ( $f''(x) = 3 > 0$ ) en todos los puntos, se concluye que:

**La startup está creciendo de forma acelerada.**

Esto significa que no solo está ganando usuarios (primera derivada positiva), sino que la *velocidad* a la que los gana (la tasa de crecimiento) aumenta de forma constante en 3,000 usuarios/mes, cada mes.

## 1.2. Código en R para Verificación

Se utiliza el siguiente script en R para comprobar los cálculos. El paquete `listings` (ya cargado en la plantilla) se usa para formatear el código.

Listing 1: Código en R para la verificación de los cálculos.

```
# =====  
# EJERCICIO 1: ANALISIS DE CRECIMIENTO DE USUARIOS  
# Diferenciacion numerica – Solucion en R  
# Datos: usuarios mensuales acumulados (en miles)  
# Fuente: PDF "Diferenciacion Numerica".  
# =====  
  
# — Datos —  
usuarios <- c(10, 15, 23, 34, 48, 65, 85) # f1 .. f7 (miles)  
h <- 1 # paso en meses  
  
# — 1) Derivadas primeras —  
# Mes 1: diferencia hacia adelante  
fwd_mes1 <- (usuarios[2] – usuarios[1]) / h  
  
# Mes 4: diferencia centrada (f(x+h)–f(x–h))/(2h) → indices: 5 y 3  
central_mes4 <- (usuarios[5] – usuarios[3]) / (2 * h)  
  
# Mes 7: diferencia hacia atras (f(x)–f(x–h))/h  
bwd_mes7 <- (usuarios[7] – usuarios[6]) / h  
  
# — 2) Segunda derivada para cada punto (3–puntos)  
# Usaremos la formula centrada para puntos internos y  
# versiones equivalentes de 3 puntos en los extremos.  
n <- length(usuarios)  
segunda <- numeric(n)  
  
# Para i = 1 (extremo izquierdo) usamos f3 – 2 f2 + f1  
segunda[1] <- usuarios[3] – 2 * usuarios[2] + usuarios[1]
```



```
# Para i = 2 .. n-1 usamos f[i+1] - 2 f[i] + f[i-1]
for (i in 2:(n-1)) {
  segunda[i] <- usuarios[i + 1] - 2 * usuarios[i] + usuarios[i - 1]
}

# Para i = n (extremo derecho) usamos f[n] - 2 f[n-1] + f[n-2]
segunda[n] <- usuarios[n] - 2 * usuarios[n - 1] + usuarios[n - 2]

# --- 3) Mostrar resultados (con unidades) ---
cat("=====\n")
cat("EJERCICIO 1 – Diferenciacion Numerica (R)\n")
cat("Datos: usuarios mensuales (miles). Paso h =", h, "mes(es)\n")
cat("-----\n")
cat(sprintf("Derivada (mes 1, adelante): %.4g miles usuarios/mes\n", fwd_1))
cat(sprintf("Derivada (mes 4, centrada): %.4g miles usuarios/mes\n", cent_4))
cat(sprintf("Derivada (mes 7, atras): %.4g miles usuarios/mes\n\n", bwd_7))

cat("Segunda derivada (aceleracion) por mes (miles usuarios / mes^2):\n")
for (i in 1:n) {
  cat(sprintf(" Mes %d: %g\n", i, segunda[i]))
}
cat("-----\n")
cat("Interpretacion:\n")
cat(" – Aceleracion constante y positiva (3 miles usuarios / mes^2).\n")
cat(" – La curva es concava hacia arriba -> crecimiento acelerado.\n")
cat("=====\n")
```

### 1.3. Salida de la Consola

La ejecución del script en R produce la siguiente salida, confirmando los resultados manuales.

Listing 2: Salida de la consola de R.

```
=====  
EJERCICIO 1 – Diferenciacion Numerica (R)  
Datos: usuarios mensuales (miles). Paso h = 1 mes(es)  
-----  
Derivada (mes 1, adelante): 5 miles usuarios/mes  
Derivada (mes 4, centrada): 12.5 miles usuarios/mes  
Derivada (mes 7, atras): 20 miles usuarios/mes  
  
Segunda derivada (aceleracion) por mes (miles usuarios / mes^2):  
Mes 1: 3  
Mes 2: 3  
Mes 3: 3  
Mes 4: 3  
Mes 5: 3  
Mes 6: 3
```



Mes 7: 3

Interpretación:

- Aceleración constante y positiva (3 miles usuarios /  $\text{mes}^2$ ).
- La curva es cóncava hacia arriba  $\rightarrow$  crecimiento acelerado.

=====

## 2. Ejercicio 2: Optimización de Función de Pérdida

Durante el entrenamiento de un modelo de Machine Learning, se registra la función de pérdida (loss) en cada época:

Cuadro 2: Función de pérdida (Loss) por Época

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

### Tareas:

1. Calcula la tasa de cambio del loss en la época 20 usando diferencia centrada con  $h = 10$ .
2. Calcula la segunda derivada en la época 30. ¿Qué indica sobre la convergencia?
3. Si la tasa de cambio del loss es menor que 0.01 por época, el entrenamiento puede detenerse. ¿En qué época se alcanza este criterio?
4. Estima el loss en la época 25 usando interpolación lineal basada en las derivadas.

### 2.1. Solución del Ejercicio 2

#### Datos Iniciales del Problema

Se definen los datos como una función  $f(x)$  donde  $x$  es la época.

- $f(0) = 2,45$
- $f(10) = 1,82$
- $f(20) = 1,35$
- $f(30) = 1,08$
- $f(40) = 0,95$
- $f(50) = 0,89$
- El paso (distancia entre épocas) es:  $h = 10$ .



### 2.1.1. Tarea 1: Tasa de cambio en la época 20 (Diferencia Centrada)

■ **Fórmula:**  $f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$

■ **Cálculo:**

$$f'(20) \approx \frac{f(30) - f(10)}{2 \cdot 10} = \frac{1,08 - 1,82}{20} = \frac{-0,74}{20} = -0,037$$

■ **Resultado:** -0.037 loss / época.

■ **Interpretación:** En la época 20, el loss está disminuyendo (valor negativo) a una tasa de 0.037 por época.

### 2.1.2. Tarea 2: Segunda derivada en la época 30

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

■ **Cálculo:**

$$\begin{aligned} f''(30) &\approx \frac{f(40) - 2f(30) + f(20)}{10^2} \\ &\approx \frac{0,95 - 2(1,08) + 1,35}{100} \\ &\approx \frac{0,95 - 2,16 + 1,35}{100} \\ &\approx \frac{0,14}{100} = 0,0014 \end{aligned}$$

■ **Resultado:** 0.0014 loss / época<sup>2</sup>.

■ **Interpretación sobre convergencia:** El valor es positivo ( $f''(30) > 0$ ). Esto indica que la curva de loss es cóncava hacia arriba, lo que significa que, aunque el loss sigue disminuyendo (la derivada  $f'$  es negativa), la *velocidad* de esa disminución se está reduciendo. El entrenamiento se está **desacelerando**; cada época aporta menos mejora que la anterior.

### 2.1.3. Tarea 3: Criterio de parada ( $|f'| < 0,01$ )

Calculamos la derivada  $f'(x)$  en cada época para encontrar cuándo su magnitud es menor que 0.01.

■ **Época 0 (Adelante):**  $f'(0) \approx \frac{f(10)-f(0)}{10} = \frac{1,82-2,45}{10} = -0,063$

■ **Época 10 (Centrada):**  $f'(10) \approx \frac{f(20)-f(0)}{2 \cdot 10} = \frac{1,35-2,45}{20} = -0,055$

■ **Época 20 (Centrada):**  $f'(20) \approx -0,037$  (ya calculado)

■ **Época 30 (Centrada):**  $f'(30) \approx \frac{f(40)-f(20)}{2 \cdot 10} = \frac{0,95-1,35}{20} = -0,020$

■ **Época 40 (Centrada):**  $f'(40) \approx \frac{f(50)-f(30)}{2 \cdot 10} = \frac{0,89-1,08}{20} = -0,0095$

■ **Época 50 (Atrás):**  $f'(50) \approx \frac{f(50)-f(40)}{10} = \frac{0,89-0,95}{10} = -0,006$



### Verificación del criterio:

- Época 30:  $|-0,020| = 0,02 \not< 0,01$
- Época 40:  $|-0,0095| = 0,0095 < 0,01$  **(Cumple)**

**Conclusión:** El criterio de parada se alcanza en la **época 40**.

### 2.1.4. Tarea 4: Estimación del loss en época 25 (Interpolación Lineal)

Usamos la aproximación lineal (Taylor de primer orden) alrededor del punto más cercano conocido, la época 20.

- **Fórmula:**  $f(x) \approx f(x_0) + (x - x_0)f'(x_0)$
- **Cálculo (con  $x = 25$  y  $x_0 = 20$ ):**

$$\begin{aligned} f(25) &\approx f(20) + (25 - 20)f'(20) \\ &\approx 1,35 + 5 \cdot (-0,037) \\ &\approx 1,35 - 0,185 \\ &\approx 1,165 \end{aligned}$$

- **Resultado:** El loss estimado en la época 25 es **1.165**.

## 2.2. Código en R para Verificación

Script en R que implementa y verifica los cálculos realizados.

Listing 3: Código en R para el Ejercicio 2 (Optimización de Loss).

```
# =====
# EJERCICIO 2: OPTIMIZACION DE LOSS (Diferenciacion Numerica)
# Lenguaje: R
# Datos: epocas y loss (ver PDF).
# =====

# --- Datos ---
epocas <- c(0, 10, 20, 30, 40, 50)
loss    <- c(2.45, 1.82, 1.35, 1.08, 0.95, 0.89)

h <- 10 # paso entre epocas (10 unidades)

# --- 1) Derivada centrada en epoca 20 ---
# f'(20) approx (f(30) - f(10)) / (2*h)
f_prime_20 <- (loss[4] - loss[2]) / (2 * h) # indices: 4->30, 2->10
# Note: loss[3] corresponds to epoch 20 itself

# --- 2) Segunda derivada en epoca 30 ---
# f''(30) approx (f(40) - 2*f(30) + f(20)) / h^2
f_second_30 <- (loss[5] - 2 * loss[4] + loss[3]) / (h^2)
```



```
# — 3) Calcular derivadas en todas las epocas (centradas cuando posible)
n <- length(epocas)
f_prime <- numeric(n)

# Epoca 0: forward (no existe epoca -10)
f_prime[1] <- (loss[2] - loss[1]) / h

# Epocas internas con centrada
for (i in 2:(n-1)) {
  # corresponde a epocas[i] con vecinos epocas[i-1], epocas[i+1]
  f_prime[i] <- (loss[i + 1] - loss[i - 1]) / (2 * h)
}

# Epoca final (backward)
f_prime[n] <- (loss[n] - loss[n - 1]) / h

# — Encuentra la primera epoca donde |f'| < 0.01 —
umbral <- 0.01
cumple <- which(abs(f_prime) < umbral)
primera_epoca_cumple <- if (length(cumple) > 0) epocas[min(cumple)] else 1

# — 4) Estimar loss en epoca 25 mediante aproximacion lineal alrededor
# f(25) approx f(20) + (25 - 20) * f'(20)
# Usamos f_prime[3] porque es la derivada en la epoca 20 (el tercer elemento)
f_est_25 <- loss[3] + (25 - 20) * f_prime[3]

# — Mostrar resultados —
cat("=====\n")
cat("EJERCICIO 2 – Diferenciacion Numerica (R)\n")
cat("-----\n")
cat(sprintf("Paso h = %d epocas\n\n", h))
cat(sprintf("1) f'(20) (centrada) = %.6g (loss/epoca)\n", f_prime_20))
cat(sprintf("2) f''(30) = %.6g (loss/epoca^2)\n\n", f_second_30))

cat("Derivadas aproximadas en cada epoca (loss/epoca):\n")
for (i in 1:n) {
  cat(sprintf(" Epoca %d: %.6g\n", epocas[i], f_prime[i]))
}
cat("\n")

if (!is.na(primera_epoca_cumple)) {
  cat(sprintf("3) Primera epoca donde |f'| < %.4g : epoca %d\n\n", umbral, primera_epoca_cumple))
} else {
  cat(sprintf("3) No se encontro epoca con |f'| < %.4g en los datos.\n\n"))
}

cat(sprintf("4) Estimacion lineal f(25) approx %.6g (loss)\n", f_est_25))
cat("-----\n")
```



```
cat("Interpretacion:\n")
cat("- f'(20) negativo indica que el loss disminuye en la epoca 20.\n")
cat("- f''(30) positivo y pequeno indica desaceleracion de la mejora.\n")
cat("-----\n")
```

### 2.3. Salida de la Consola

La ejecución del script en R produce la siguiente salida, confirmando los resultados manuales.

Listing 4: Salida de la consola de R para el Ejercicio 2.

```
=====
EJERCICIO 2 – Diferenciacion Numerica (R)
-----
Paso h = 10 epocas

1) f'(20) (centrada) = -0.037 (loss/epoca)
2) f''(30) = 0.0014 (loss/epoca^2)

Derivadas aproximadas en cada epoca (loss/epoca):
Epoca 0: -0.063
Epoca 10: -0.055
Epoca 20: -0.037
Epoca 30: -0.02
Epoca 40: -0.0095
Epoca 50: -0.006

3) Primera epoca donde |f'| < 0.01 : epoca 40
4) Estimacion lineal f(25) approx 1.165 (loss)
-----
Interpretacion:
- f'(20) negativo indica que el loss disminuye en la epoca 20.
- f''(30) positivo y pequeno indica desaceleracion de la mejora.
-----
```

## 3. Ejercicio 3: Análisis de Series Temporales de Ventas

Una empresa de e-commerce registra sus ventas diarias (en miles de dólares) durante una semana de campaña:

Cuadro 3: Ventas diarias de campaña (en miles de dólares)

Día	Lun	Mar	Mié	Jue	Vie	Sáb	Dom
Ventas (\$k)	45	52	61	58	73	89	95



## Tareas:

1. Calcula la velocidad de crecimiento de ventas (derivada) para cada día usando diferencias finitas apropiadas.
2. Identifica el día con mayor aceleración de ventas (segunda derivada positiva máxima).
3. El jueves hubo una caída en la tendencia. Calcula la magnitud de esta desaceleración.
4. Si la tendencia del domingo se mantiene, ¿cuántas ventas esperarías el lunes siguiente? (usa extrapolación lineal con la derivada).

## 3.1. Solución del Ejercicio 3

### Datos Iniciales del Problema

Se definen los datos como  $f_i$  donde  $i$  es el índice del día (1=Lun, ..., 7=Dom).

- $f_1 = 45, f_2 = 52, f_3 = 61, f_4 = 58, f_5 = 73, f_6 = 89, f_7 = 95.$
- El paso (distancia entre puntos) es:  $h = 1$  día.

#### 3.1.1. Tarea 1: Velocidad de Crecimiento (Primera Derivada)

Se usa diferencia centrada para puntos internos (Mar-Sáb), diferencia hacia adelante para el Lunes y hacia atrás para el Domingo. (Unidades: \$k/día).

**Lunes (día 1) – Adelante:**  $f'(1) \approx \frac{f_2 - f_1}{h} = \frac{52 - 45}{1} = 7$

**Martes (día 2) – Centrada:**  $f'(2) \approx \frac{f_3 - f_1}{2h} = \frac{61 - 45}{2} = \frac{16}{2} = 8$

**Miércoles (día 3) – Centrada:**  $f'(3) \approx \frac{f_4 - f_2}{2h} = \frac{58 - 52}{2} = \frac{6}{2} = 3$

**Jueves (día 4) – Centrada:**  $f'(4) \approx \frac{f_5 - f_3}{2h} = \frac{73 - 61}{2} = \frac{12}{2} = 6$

**Viernes (día 5) – Centrada:**  $f'(5) \approx \frac{f_6 - f_4}{2h} = \frac{89 - 58}{2} = \frac{31}{2} = 15,5$

**Sábado (día 6) – Centrada:**  $f'(6) \approx \frac{f_7 - f_5}{2h} = \frac{95 - 73}{2} = \frac{22}{2} = 11$

**Domingo (día 7) – Atrás:**  $f'(7) \approx \frac{f_7 - f_6}{h} = \frac{95 - 89}{1} = 6$

**Resultados (Velocidad):** Las velocidades (\$k/día) de Lun a Dom son [7, 8, 3, 6, 15.5, 11, 6].



### 3.1.2. Tarea 2: Día con Mayor Aceleración (Segunda Derivada)

Se calcula la segunda derivada (aceleración) para los puntos internos (días 2 a 6) usando la fórmula  $f''(i) \approx f_{i+1} - 2f_i + f_{i-1}$  (ya que  $h = 1$ ). (Unidades: \$k/día<sup>2</sup>).

- **Martes (día 2):**  $f''(2) = f_3 - 2f_2 + f_1 = 61 - 2(52) + 45 = 61 - 104 + 45 = 2$
- **Miércoles (día 3):**  $f''(3) = f_4 - 2f_3 + f_2 = 58 - 2(61) + 52 = 58 - 122 + 52 = -12$
- **Jueves (día 4):**  $f''(4) = f_5 - 2f_4 + f_3 = 73 - 2(58) + 61 = 73 - 116 + 61 = 18$
- **Viernes (día 5):**  $f''(5) = f_6 - 2f_5 + f_4 = 89 - 2(73) + 58 = 89 - 146 + 58 = 1$
- **Sábado (día 6):**  $f''(6) = f_7 - 2f_6 + f_5 = 95 - 2(89) + 73 = 95 - 178 + 73 = -10$

**Resultado:** El día con la mayor aceleración positiva es el **Jueves (día 4)** con un valor de **18 \$k/día<sup>2</sup>**.

### 3.1.3. Tarea 3: Magnitud de la Caída el Jueves

La caída en la tendencia se refiere a la disminución neta de ventas entre el Miércoles y el Jueves.

- **Cálculo:**  $\Delta \text{Ventas} = f(\text{Jue}) - f(\text{Mié}) = f_4 - f_3 = 58 - 61 = -3$  (\$k)
- **Resultado:** La magnitud de la caída fue de **3 \$k** (es decir, una disminución de 3,000 dólares).

### 3.1.4. Tarea 4: Estimación Ventas Lunes Siguiente

Usamos la extrapolación lineal (Taylor de 1er orden) basada en los datos del Domingo (día 7).

- **Fórmula:**  $f(x + h) \approx f(x) + h \cdot f'(x)$
- **Datos:**  $f(7) = 95$  y  $f'(7) = 6$  (calculado en Tarea 1).  $h = 1$ .
- **Cálculo:**  
$$\text{Venta Lunes} \approx f(7) + 1 \cdot f'(7) = 95 + 1 \cdot 6 = 101$$
- **Resultado:** Se esperarían ventas de **101 \$k** el Lunes siguiente.

## 3.2. Código en R para Verificación

Script en R que implementa y verifica los cálculos.

Listing 5: Código en R para el Ejercicio 3 (Análisis de Ventas).

```
# =====
# EJERCICIO 3: ANALISIS DE VENTAS DIARIAS
# Diferenciacion Numerica – Solucion en R
# Datos: ventas diarias (en miles de $)
# Fuente: PDF "Diferenciacion Numerica".
# =====
```



```
# --- Datos ---
dias <- c("Lun", "Mar", "Mie", "Jue", "Vie", "Sab", "Dom")
ventas <- c(45, 52, 61, 58, 73, 89, 95) # en $k
h <- 1 # paso en dias

# --- 1) Primera derivada (velocidad) ---
n <- length(ventas)
velocidad <- numeric(n)

# Lunes -> forward
velocidad[1] <- (ventas[2] - ventas[1]) / h

# Dias internos -> central
for (i in 2:(n-1)) {
  velocidad[i] <- (ventas[i + 1] - ventas[i - 1]) / (2 * h)
}

# Domingo -> backward
velocidad[n] <- (ventas[n] - ventas[n - 1]) / h

# --- 2) Segunda derivada (aceleracion) para i = 2..n-1 ---
aceleracion <- rep(NA, n)
for (i in 2:(n-1)) {
  aceleracion[i] <- ventas[i + 1] - 2 * ventas[i] + ventas[i - 1]
# h=1 simplifica
}

# --- 3) Dia con mayor aceleracion positiva ---
# Buscamos el maximo de aceleracion entre los valores no-NA
max_acc_value <- max(aceleracion, na.rm = TRUE)
max_acc_index <- which(aceleracion == max_acc_value)
dia_max_acc <- dias[max_acc_index]

# --- 4) Magnitud de la caida entre miercoles y jueves ---
caida_wed_thu <- ventas[4] - ventas[3] # Jue - Mie (puede ser negativo)

# --- 5) Extrapolacion lineal: ventas lunes siguiente usando derivada de
ventas_lunes_sig <- ventas[n] + velocidad[n] * 1 # f(7) + f'(7)*1

# --- Mostrar resultados ---
cat("=====\n")
cat("EJERCICIO 3 - Ventas diarias (Diferenciacion Numerica)\n")
cat("-----\n")
cat("Datos (ventas, en miles $):\n")
for (i in 1:n) cat(sprintf(" %s: %g\n", dias[i], ventas[i]))
cat("\nPrimera derivada (velocidad, $k/dia):\n")
for (i in 1:n) cat(sprintf(" %s: %g\n", dias[i], velocidad[i]))
cat("\nSegunda derivada (aceleracion, $k/dia^2) (NA en extremos):\n")
```



```
for (i in 1:n) cat(sprintf(" %s: %s\n", dias[i], ifelse(is.na(acceleracion
cat("\nDia(s) con mayor aceleracion positiva:\n")
cat(sprintf(" %s con aceleracion = %g $k/dia^2\n\n", paste(dia_max_acc, c
cat(sprintf("Caida Mie -> Jue (Jue - Mie) = %g $k (negativo indica dismin
cat(sprintf("Estimacion ventas lunes siguiente = %g $k\n", ventas_lunes_s
cat("-----\n")
cat("Interpretacion:\n")
cat("- Jueves muestra la mayor aceleracion positiva (fuerte aumento en la
cat("- Aunque hubo una caida en el nivel Mie->Jue de 3 $k, la dinamica en
cat("=====\n")
```

### 3.3. Salida de la Consola

La ejecución del script en R produce la siguiente salida, confirmando los resultados manuales.

Listing 6: Salida de la consola de R para el Ejercicio 3.

```
=====  
EJERCICIO 3 – Ventas diarias (Diferenciacion Numerica)  
-----  
Datos (ventas, en miles $):  
Lun: 45  
Mar: 52  
Mie: 61  
Jue: 58  
Vie: 73  
Sab: 89  
Dom: 95  
  
Primera derivada (velocidad, $k/dia):  
Lun: 7  
Mar: 8  
Mie: 3  
Jue: 6  
Vie: 15.5  
Sab: 11  
Dom: 6  
  
Segunda derivada (aceleracion, $k/dia^2) (NA en extremos):  
Lun: NA  
Mar: 2  
Mie: -12  
Jue: 18  
Vie: 1  
Sab: -10  
Dom: NA
```



Día(s) con mayor aceleración positiva:

Jue con aceleración = 18 \$k/día<sup>2</sup>

Caida Mie → Jue (Jue – Mie) = -3 \$k (negativo indica disminución)

Estimación ventas lunes siguiente = 101 \$k

Interpretación:

- Jueves muestra la mayor aceleración positiva (fuerte aumento en la velocidad)
- Aunque hubo una caída en el nivel Mie → Jue de 3 \$k, la dinámica entre días

=====

## 4. Ejercicio 4: Gradiente de Función de Activación

En redes neuronales, necesitas calcular el gradiente de la función sigmoide  $\sigma(x) = \frac{1}{1+e^{-x}}$  en varios puntos para el backpropagation. No tienes la derivada analítica programada. Datos evaluados de la función:

Cuadro 4: Valores de la función Sigmoide  $\sigma(x)$

$x$	-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

**Tareas:**

1. Calcula  $\sigma'(0)$  usando diferencia centrada con  $h = 1$ .
2. Calcula  $\sigma'(-2)$  y  $\sigma'(2)$  usando diferencias centradas (con  $h = 1$ ).
3. Compara tus resultados numéricos con la derivada analítica:  $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ .
4. ¿Qué tamaño de  $h$  recomendarías para este cálculo? Justifica.
5. ¿Por qué la derivada es simétrica alrededor de  $x = 0$ ?

### 4.1. Solución del Ejercicio 4

Se utiliza la fórmula de diferencia centrada:  $\sigma'(x) \approx \frac{\sigma(x+h) - \sigma(x-h)}{2h}$ . Con  $h = 1$ :

#### 4.1.1. Tarea 1: Cálculo de $\sigma'(0)$

$$\sigma'(0) \approx \frac{\sigma(1) - \sigma(-1)}{2 \cdot 1} = \frac{0,7311 - 0,2689}{2} = \frac{0,4622}{2}$$

**Resultado Numérico:**  $\sigma'(0) \approx 0,2311$



#### 4.1.2. Tarea 2: Cálculo de $\sigma'(-2)$ y $\sigma'(2)$

Cálculo de  $\sigma'(-2)$ :

$$\sigma'(-2) \approx \frac{\sigma(-1) - \sigma(-3)}{2 \cdot 1} = \frac{0,2689 - 0,0474}{2} = \frac{0,2215}{2}$$

**Resultado Numérico:**  $\sigma'(-2) \approx 0,11075$

Cálculo de  $\sigma'(2)$ :

$$\sigma'(2) \approx \frac{\sigma(3) - \sigma(1)}{2 \cdot 1} = \frac{0,9526 - 0,7311}{2} = \frac{0,2215}{2}$$

**Resultado Numérico:**  $\sigma'(2) \approx 0,11075$

#### 4.1.3. Tarea 3: Comparación con la Derivada Analítica

La derivada analítica es  $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ .

En  $x = 0$ :

- **Analítico:**  $\sigma'(0) = \sigma(0)(1 - \sigma(0)) = 0,5000 \times (1 - 0,5000) = 0,2500$
- **Numérico ( $h = 1$ ):** 0,2311
- **Error (Num - Ana):**  $0,2311 - 0,2500 = -0,0189$

En  $x = 2$ :

- **Analítico:**  $\sigma'(2) = \sigma(2)(1 - \sigma(2)) = 0,8808 \times (1 - 0,8808) = 0,8808 \times 0,1192 \approx 0,10499$
- **Numérico ( $h = 1$ ):** 0,11075
- **Error (Num - Ana):**  $0,11075 - 0,10499 = +0,00576$

**Observación:** El error es relativamente grande porque  $h = 1$  es un paso muy grande para esta función, introduciendo un error de truncamiento significativo.

#### 4.1.4. Tarea 4: Recomendación del tamaño de $h$

Un  $h = 1$  es demasiado grande y produce un error de truncamiento notable.

- **Error de Truncamiento:** La fórmula de diferencia centrada tiene un error del orden  $O(h^2)$ . Reducir  $h$  (p.ej., a 0.1) reduce este error cuadráticamente.
- **Error de Redondeo:** Si  $h$  se hace demasiado pequeño (p.ej.,  $h < 10^{-8}$ ), la resta  $\sigma(x+h) - \sigma(x-h)$  sufre de "cancelación catastrófica" (error de redondeo), y este error se magnifica al dividir por un  $h$  diminuto.
- **Recomendación:** Un balance óptimo entre estos dos errores para la diferencia centrada en precisión doble (64 bits) se encuentra típicamente alrededor de  $h \approx 10^{-5}$  a  $h \approx 10^{-6}$ .



#### 4.1.5. Tarea 5: Simetría de la Derivada

La derivada  $\sigma'(x)$  es simétrica (una función par) porque la propia función sigmoide  $\sigma(x)$  tiene una propiedad de simetría rotacional alrededor del punto  $(0, 0,5)$ .

**Prueba analítica:** La función sigmoide cumple la propiedad:  $\sigma(-x) = 1 - \sigma(x)$ . Si derivamos ambos lados de la ecuación usando la regla de la cadena:

$$\begin{aligned} \frac{d}{dx}[\sigma(-x)] &= \frac{d}{dx}[1 - \sigma(x)] \\ \sigma'(-x) \cdot \frac{d}{dx}[-x] &= 0 - \sigma'(x) \\ \sigma'(-x) \cdot (-1) &= -\sigma'(x) \\ -\sigma'(-x) &= -\sigma'(x) \\ \sigma'(-x) &= \sigma'(x) \end{aligned}$$

Una función que cumple  $f'(-x) = f'(x)$  es, por definición, una función par (simétrica respecto al eje Y). Esto explica por qué  $\sigma'(-2)$  y  $\sigma'(2)$  dieron el mismo resultado.

## 4.2. Código en R para Verificación

Script en R que implementa y verifica los cálculos.

Listing 7: Código en R para el Ejercicio 4 (Gradiente Sigmoide).

```
# =====
# EJERCICIO 4: Gradiente de la sigmoide (Diferenciacion Numerica)
# Lenguaje: R
# Datos: valores de sigma(x) en x = -3, -2, -1, 0, 1, 2, 3 (ver PDF)
# =====

# — Datos (tal cual del PDF) —
x_vals <- c(-3, -2, -1, 0, 1, 2, 3)
sigma_vals <- c(0.0474, 0.1192, 0.2689, 0.5000, 0.7311, 0.8808, 0.9526)
names(sigma_vals) <- as.character(x_vals)

# — Funciones auxiliares —
centered_diff <- function(x, sigma_map, h=1) {
  # sigma_map: named vector with sigma values indexed by x
  x_minus <- as.character(x - h)
  x_plus <- as.character(x + h)
  (sigma_map[x_plus] - sigma_map[x_minus]) / (2 * h)
}

analytic_derivative <- function(sigma_x) {
  sigma_x * (1 - sigma_x)
}

# — 1) Derivadas centradas con h = 1 (tal como pide el PDF) —
```



```
h1 <- 1
# puntos de interes: -2, 0, 2
points <- c(-2, 0, 2)
num_derivs_h1 <- sapply(points, function(p) centered_diff(p, sigma_vals, h1))
names(num_derivs_h1) <- as.character(points)

# — 2) Derivadas analiticas en esos puntos —
analytic_vals <- sapply(points, function(p) analytic_derivative(sigma_val, h1, p))
names(analytic_vals) <- as.character(points)

# — 3) Errores numerico - analitico —
errors_h1 <- num_derivs_h1 - analytic_vals

# — 4) Opcional: ver como mejora con h pequeno (ej. h = 1e-5) —
h_small <- 1e-5
# Por demostracion, si tuvieras la funcion sigmoide:
sigmoid <- function(x) 1 / (1 + exp(-x))
num_derivs_hsmall <- sapply(points, function(p) {
  (sigmoid(p + h_small) - sigmoid(p - h_small)) / (2 * h_small)
})
names(num_derivs_hsmall) <- as.character(points)

# — Mostrar resultados —
cat("=====\n")
cat("EJERCICIO 4 – Gradiente de la sigmoide (R)\n")
cat("Datos: sigma(-3..3) tal cual en el PDF\n")
cat("-----\n")
cat("Derivadas numericas (centradas) con h = 1 (Pedido en Tareas 1 y 2):\n")
for (p in points) {
  cat(sprintf(" x = %3s : num(h=1) = %.6g   analitica = %.8g   (error = %10g)\n",
             p, num_derivs_h1[as.character(p)], analytic_vals[as.character(p)], errors_h1[as.character(p)]))
}
cat("\nNota: los valores numericos con h=1 corresponden exactamente al calculo analitico\n")

cat("\nDemostracion (Tarea 4) de mejora con h = 1e-5 (usando la funcion sigmoide)\n")
for (p in points) {
  cat(sprintf(" x = %3s : num(h=1e-5) = %.10g   analitica = %.10g   (diff = %10g)\n",
             p, num_derivs_hsmall[as.character(p)], analytic_vals[as.character(p)], errors_h1[as.character(p)]))
}
cat("-----\n")
cat("Recomendacion (Tarea 4): usar h approx 1e-5 para balancear truncamiento y error de redondeo\n")
cat("Tarea 5 (Simetria): La derivada es una funcion par, sigma'(-x) = sigma'(x)\n")
cat("=====\n")
```

### 4.3. Salida de la Consola

La ejecución del script en R produce la siguiente salida.



Listing 8: Salida de la consola de R para el Ejercicio 4.

=====

EJERCICIO 4 – Gradiente de la sigmoide (R)

Datos: sigma(-3..3) tal cual en el PDF

Derivadas numericas (centradas) con h = 1 (Pedido en Tareas 1 y 2):

```
x = -2 : num(h=1) = 0.11075    analitica = 0.10499136    (error = 0.00575)
x =  0 : num(h=1) = 0.2311     analitica = 0.25              (error = -0.0189)
x =  2 : num(h=1) = 0.11075    analitica = 0.10499136    (error = 0.00575)
```

Nota: los valores numericos con h=1 corresponden exactamente al calculo p

Demostracion (Tarea 4) de mejora con h = 1e-5 (usando la funcion sigmoide

```
x = -2 : num(h=1e-5) = 0.10499359    analitica = 0.10499136    (diff = 2.2)
x =  0 : num(h=1e-5) = 0.25           analitica = 0.25          (diff = 4.1)
x =  2 : num(h=1e-5) = 0.10499359    analitica = 0.10499136    (diff = 2.2)
```

Recomendacion (Tarea 4): usar h approx 1e-5 para balancear truncamiento/r  
Tarea 5 (Simetria): La derivada es una funcion par,  $\sigma'(-x) = \sigma'(x)$

=====

## 5. Ejercicio 5: Detección de Anomalías en Métricas de Sistema

Un sistema de monitoreo registra el tiempo de respuesta (en ms) de una API cada hora durante un incidente:

Cuadro 5: Latencia de API por hora (en ms)

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

### Tareas:

1. Calcula la primera derivada (tasa de cambio) para cada hora usando diferencias finitas apropiadas.
2. Identifica el momento del "pico de anomalía" calculando dónde la segunda derivada cambia de signo (de positiva a negativa).
3. Entre las horas 3 y 4 hay un salto brusco. Calcula la magnitud de este cambio.
4. A partir de la hora 6, el sistema comienza a recuperarse. Calcula la tasa de recuperación (derivada negativa).
5. Si defines una "anomalía" como un cambio mayor a 50 ms/hora, ¿en qué momentos se detectarían anomalías?



## 5.1. Solución del Ejercicio 5

### Datos Iniciales del Problema

Se definen los datos como  $f_i$  donde  $i$  es la hora.

- $f_0 = 120, f_1 = 125, f_2 = 128, f_3 = 135, f_4 = 280, f_5 = 290, f_6 = 275, f_7 = 155.$
- El paso (distancia entre puntos) es:  $h = 1$  hora.

#### 5.1.1. Tarea 1: Primera Derivada (Tasa de Cambio)

Se usa diferencia hacia adelante en la hora 0, hacia atrás en la hora 7, y centrada en las horas 1-6. (Unidades: ms/h).

- **Hora 0 (Adelante):**  $f'(0) \approx \frac{f_1 - f_0}{h} = \frac{125 - 120}{1} = 5$
- **Hora 1 (Centrada):**  $f'(1) \approx \frac{f_2 - f_0}{2h} = \frac{128 - 120}{2} = \frac{8}{2} = 4$
- **Hora 2 (Centrada):**  $f'(2) \approx \frac{f_3 - f_1}{2h} = \frac{135 - 125}{2} = \frac{10}{2} = 5$
- **Hora 3 (Centrada):**  $f'(3) \approx \frac{f_4 - f_2}{2h} = \frac{280 - 128}{2} = \frac{152}{2} = 76$
- **Hora 4 (Centrada):**  $f'(4) \approx \frac{f_5 - f_3}{2h} = \frac{290 - 135}{2} = \frac{155}{2} = 77,5$
- **Hora 5 (Centrada):**  $f'(5) \approx \frac{f_6 - f_4}{2h} = \frac{275 - 280}{2} = \frac{-5}{2} = -2,5$
- **Hora 6 (Centrada):**  $f'(6) \approx \frac{f_7 - f_5}{2h} = \frac{155 - 290}{2} = \frac{-135}{2} = -67,5$
- **Hora 7 (Atrás):**  $f'(7) \approx \frac{f_7 - f_6}{h} = \frac{155 - 275}{1} = -120$

**Resultados (Tasa de Cambio):** Las tasas (ms/h) de la hora 0 a la 7 son: [5, 4, 5, 76, 77.5, -2.5, -67.5, -120].

#### 5.1.2. Tarea 2: Pico de Anomalía (Segunda Derivada)

Se calcula la segunda derivada (aceleración) para los puntos internos (horas 1 a 6) usando  $f''(i) \approx f_{i+1} - 2f_i + f_{i-1}$  (ya que  $h = 1$ ). (Unidades: ms/h<sup>2</sup>).

- **Hora 1:**  $f''(1) = f_2 - 2f_1 + f_0 = 128 - 2(125) + 120 = 128 - 250 + 120 = -2$
- **Hora 2:**  $f''(2) = f_3 - 2f_2 + f_1 = 135 - 2(128) + 125 = 135 - 256 + 125 = 4$
- **Hora 3:**  $f''(3) = f_4 - 2f_3 + f_2 = 280 - 2(135) + 128 = 280 - 270 + 128 = 138$
- **Hora 4:**  $f''(4) = f_5 - 2f_4 + f_3 = 290 - 2(280) + 135 = 290 - 560 + 135 = -135$
- **Hora 5:**  $f''(5) = f_6 - 2f_5 + f_4 = 275 - 2(290) + 280 = 275 - 580 + 280 = -25$
- **Hora 6:**  $f''(6) = f_7 - 2f_6 + f_5 = 155 - 2(275) + 290 = 155 - 550 + 290 = -105$

**Resultado:** El "pico de anomalía" ocurre donde la segunda derivada cambia de signo positivo a negativo. Esto sucede entre la **Hora 3** ( $f''(3) = +138$ ) y la **Hora 4** ( $f''(4) = -135$ ). El punto máximo de latencia (290 ms) se da en la Hora 5, pero el cambio más brusco (el pico de la anomalía) ocurre al llegar a la Hora 4.



### 5.1.3. Tarea 3: Magnitud del Salto (Horas 3 y 4)

Se calcula la diferencia directa en los valores de latencia.

- **Cálculo:**  $\Delta = f(4) - f(3) = 280 - 135 = 145$  ms
- **Resultado:** La magnitud del salto brusco fue de **145 ms**.

### 5.1.4. Tarea 4: Tasa de Recuperación (a partir Hora 6)

La tasa de recuperación es la derivada (negativa) en ese punto.

- **Cálculo:** Se toma el valor de  $f'(6)$  calculado en la Tarea 1.
- **Resultado:** La tasa de recuperación en la hora 6 es de **-67.5 ms/h** (la latencia está disminuyendo rápidamente).

### 5.1.5. Tarea 5: Detección de Anomalías (Umbral > 50 ms/h)

Se revisa la magnitud (valor absoluto) de la primera derivada (calculada en Tarea 1) para cada hora, contra el umbral de 50 ms/h.

Cuadro 6: Verificación de Umbral de Anomalía ( $|f'| > 50$  ms/h)

Hora ( $i$ )	$ f'(i) $ (ms/h)	¿Anomalía? (> 50)
0	5.0	No
1	4.0	No
2	5.0	No
3	<b>76.0</b>	<b>Sí</b>
4	<b>77.5</b>	<b>Sí</b>
5	2.5	No
6	<b>67.5</b>	<b>Sí</b>
7	<b>120.0</b>	<b>Sí</b>

**Resultado:** Se detectarían anomalías en las horas **3, 4, 6 y 7**.

## 5.2. Código en R para Verificación

Script en R que implementa y verifica los cálculos.

Listing 9: Código en R para el Ejercicio 5 (Detección de Anomalías).

```
# =====
# EJERCICIO 5: DETECCION DE ANOMALIAS EN LATENCIAS
# Diferenciacion Numerica – Solucion en R
# Datos: latencia por hora (ms)
# Fuente: PDF "Diferenciacion Numerica".
# =====

# —— Datos ——
```



```
horas <- 0:7
latencia <- c(120, 125, 128, 135, 280, 290, 275, 155) # ms
h <- 1 # paso en horas

# --- 1) Primera derivada (tasa de cambio) ---
n <- length(latencia)
f_prime <- numeric(n)

# Hora 0 (forward)
f_prime[1] <- (latencia[2] - latencia[1]) / h

# Horas internas (central)
for (i in 2:(n-1)) {
  f_prime[i] <- (latencia[i + 1] - latencia[i - 1]) / (2 * h)
}

# Hora 7 (backward)
f_prime[n] <- (latencia[n] - latencia[n - 1]) / h

# --- 2) Segunda derivada (3-puntos) para i = 2..n-1 (horas 1..6) ---
f_second <- rep(NA, n)
for (i in 2:(n-1)) {
  f_second[i] <- latencia[i + 1] - 2 * latencia[i] + latencia[i - 1]
# h=1 simplifica
}

# --- 3) Detectar pico de anomalia: buscar cambio de signo + -> - en f_se
# (Indices 1-based en R: f_second[4] es hora 3, f_second[5] es hora 4)
pico_indices <- which((f_second[1:(n-2)] > 0) & (f_second[2:(n-1)] < 0))
pico_horas <- if (length(pico_indices) > 0) horas[pico_indices] else NA

# --- 4) Magnitud del salto entre hora 3 y 4 ---
# (Indices 1-based en R: latencia[4] es hora 3, latencia[5] es hora 4)
salto_3_4 <- latencia[5] - latencia[4]

# --- 5) Tasa de recuperacion desde hora 6 (derivada en hora 6) ---
# (Indices 1-based en R: f_prime[7] es derivada en hora 6)
tasa_recuperacion_h6 <- f_prime[7]

# --- 6) Deteccion de anomalias con umbral |Delta| > 50 ms/h ---
umbral <- 50
anom_indices <- which(abs(f_prime) > umbral)
anom_horas <- horas[anom_indices]

# --- Mostrar resultados ---
cat("=====\n")
cat("EJERCICIO 5 - Deteccion de anomalias en latencia (R)\n")
cat("=====\n")
```



```
cat("Datos (latencia en ms) por hora:\n")
for (i in 1:n) cat(sprintf(" Hora %d: %g ms\n", horas[i], latencia[i]))
cat("\nPrimera derivada (tasa de cambio, ms/h):\n")
for (i in 1:n) cat(sprintf(" Hora %d: %g\n", horas[i], f_prime[i]))
cat("\nSegunda derivada (ms/h^2) (NA en extremos):\n")
for (i in 1:n) cat(sprintf(" Hora %d: %s\n", horas[i], ifelse(is.na(f_secundaria[i]), "NA", f_secundaria[i])))

cat("\nPico(s) de anomalia (cambio + -> - en f'') detectado en hora: ", i)
cat(sprintf("Salto entre hora 3 y 4: %g ms\n", salto_3_4))
cat(sprintf("Tasa de recuperacion aprox. en hora 6: %g ms/h (negativa = recuperacion)\n", f_prime[6]))
cat(sprintf("Anomalias detectadas con |Delta| > %g ms/h en horas: %s\n", horas[anomalias]))
cat("=====\n")
```

### 5.3. Salida de la Consola

La ejecución del script en R produce la siguiente salida.

Listing 10: Salida de la consola de R para el Ejercicio 5.

```
=====  
EJERCICIO 5 – Deteccion de anomalias en latencia (R)
```

---

Datos (latencia en ms) por hora:

```
Hora 0: 120 ms  
Hora 1: 125 ms  
Hora 2: 128 ms  
Hora 3: 135 ms  
Hora 4: 280 ms  
Hora 5: 290 ms  
Hora 6: 275 ms  
Hora 7: 155 ms
```

Primera derivada (tasa de cambio, ms/h):

```
Hora 0: 5  
Hora 1: 4  
Hora 2: 5  
Hora 3: 76  
Hora 4: 77.5  
Hora 5: -2.5  
Hora 6: -67.5  
Hora 7: -120
```

Segunda derivada (ms/h^2) (NA en extremos):

```
Hora 0: NA  
Hora 1: -2  
Hora 2: 4  
Hora 3: 138  
Hora 4: -135  
Hora 5: -25
```



Hora 6: -105

Hora 7: NA

Pico(s) de anomalía (cambio + → - en f'') detectado en hora: 4

Salto entre hora 3 y 4: 145 ms

Tasa de recuperación aprox. en hora 6: -67.5 ms/h (negativa = recuperación)

Anomalías detectadas con  $|\Delta| > 50$  ms/h en horas: 3, 4, 6, 7

=====

## 6. Ejercicio 6: Análisis de Tasa de Conversión

Una campaña de marketing muestra la siguiente tasa de conversión (porcentaje) en función del gasto en publicidad (en miles de dólares):

Cuadro 7: Tasa de Conversión (%) vs. Gasto (\$k)

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

### Tareas:

1. Calcula el ROI marginal (derivada de conversión respecto al gasto) en cada punto usando diferencias centradas (y unilaterales en extremos).
2. ¿En qué rango de gasto el ROI marginal es mayor que 0.2% por cada \$1000 invertido?
3. La segunda derivada indica rendimientos decrecientes. Calcula la segunda derivada en \$15k.
4. Con base en las derivadas, ¿recomendarías aumentar el gasto más allá de \$25k? Justifica matemáticamente.

### 6.1. Solución del Ejercicio 6

#### Datos Iniciales del Problema

Se definen los datos como  $c(g)$  donde  $g$  es el gasto en \$k y  $c$  es la conversión en %.

- $c(0) = 2,1$ ,  $c(5) = 3,8$ ,  $c(10) = 5,2$ ,  $c(15) = 6,1$ ,  $c(20) = 6,7$ ,  $c(25) = 7,0$ .
- El paso (distancia entre puntos) es:  $h = 5$  (miles de dólares).

Las unidades del ROI marginal (primera derivada) serán %/\$k.

#### 6.1.1. Tarea 1: ROI Marginal (Primera Derivada)

Se usa diferencia hacia adelante en \$0k, hacia atrás en \$25k, y centrada en los puntos internos.



**Gasto \$0k (Adelante):**

$$f'(0) \approx \frac{c(5) - c(0)}{h} = \frac{3,8 - 2,1}{5} = \frac{1,7}{5} = 0,34$$

**Gasto \$5k (Centrada):**

$$f'(5) \approx \frac{c(10) - c(0)}{2h} = \frac{5,2 - 2,1}{2 \cdot 5} = \frac{3,1}{10} = 0,31$$

**Gasto \$10k (Centrada):**

$$f'(10) \approx \frac{c(15) - c(5)}{2h} = \frac{6,1 - 3,8}{10} = \frac{2,3}{10} = 0,23$$

**Gasto \$15k (Centrada):**

$$f'(15) \approx \frac{c(20) - c(10)}{2h} = \frac{6,7 - 5,2}{10} = \frac{1,5}{10} = 0,15$$

**Gasto \$20k (Centrada):**

$$f'(20) \approx \frac{c(25) - c(15)}{2h} = \frac{7,0 - 6,1}{10} = \frac{0,9}{10} = 0,09$$

**Gasto \$25k (Atrás):**

$$f'(25) \approx \frac{c(25) - c(20)}{h} = \frac{7,0 - 6,7}{5} = \frac{0,3}{5} = 0,06$$

**Resultados (ROI marginal %/\$k):** Para los gastos [0, 5, 10, 15, 20, 25] \$k, los ROI son [0.34, 0.31, 0.23, 0.15, 0.09, 0.06].

**6.1.2. Tarea 2: Rango de Gasto con ROI >0.2 %/\$k**

Comparamos los valores de la Tarea 1 con el umbral de 0.2:

- Gasto \$0k:  $0,34 > 0,2$  **(Cumple)**
- Gasto \$5k:  $0,31 > 0,2$  **(Cumple)**
- Gasto \$10k:  $0,23 > 0,2$  **(Cumple)**
- Gasto \$15k:  $0,15 \not> 0,2$  (No cumple)
- Gasto \$20k:  $0,09 \not> 0,2$  (No cumple)
- Gasto \$25k:  $0,06 \not> 0,2$  (No cumple)

**Conclusión:** El ROI marginal es mayor que 0.2 %/\$k en el rango de gasto de **\$0k a \$10k**.



### 6.1.3. Tarea 3: Segunda Derivada en \$15k

Usamos la fórmula de 3 puntos centrada con  $h = 5$ , por lo tanto  $h^2 = 25$ .

$$\begin{aligned} f''(15) &\approx \frac{c(20) - 2c(15) + c(10)}{h^2} \\ &\approx \frac{6,7 - 2(6,1) + 5,2}{25} \\ &\approx \frac{6,7 - 12,2 + 5,2}{25} \\ &\approx \frac{-0,3}{25} = -0,012 \end{aligned}$$

**Resultado:**  $f''(15) \approx -0,012 \%/(\$k)^2$ . El signo negativo confirma los **rendimientos decrecientes** (la tasa de crecimiento está disminuyendo).

### 6.1.4. Tarea 4: Recomendación sobre Aumentar Gasto

**No se recomendaría aumentar el gasto más allá de \$25k.**

**Justificación Matemática:**

1. **ROI Marginal Bajo:** La primera derivada en \$25k (calculada con diferencia hacia atrás) es  $f'(25) \approx 0,06 \%/ \$k$ . Esto es extremadamente bajo, significando que cada \$1000 adicionales invertidos solo retornan un 0.06 % de conversión extra.
2. **Rendimientos Decrecientes:** La segunda derivada en \$15k (y en puntos cercanos) es negativa ( $f''(15) \approx -0,012$ ), lo que indica que la curva es cóncava hacia abajo. Esto confirma que el ROI marginal está en una tendencia decreciente.

La inversión ha llegado a un punto de saturación.

## 6.2. Código en R para Verificación

Script en R que implementa y verifica los cálculos.

Listing 11: Código en R para el Ejercicio 6 (ROI Marginal).

```
# =====
# EJERCICIO 6: TASA DE CONVERSION VS GASTO (ROI MARGINAL)
# Diferenciacion Numerica – Solucion en R
# Datos: gasto ($k) y conversion (%) (ver PDF).
# =====

# — Datos —
gasto_k <- c(0, 5, 10, 15, 20, 25)          # en $k
conversion <- c(2.1, 3.8, 5.2, 6.1, 6.7, 7.0) # en %

h <- 5 # paso en $k
```



```
# — 1) ROI marginal (derivada) en cada punto —
n <- length(gasto_k)
roi <- numeric(n)

# Punto 0: forward
roi[1] <- (conversion[2] - conversion[1]) / h

# Puntos internos: centrada
for (i in 2:(n-1)) {
  roi[i] <- (conversion[i + 1] - conversion[i - 1]) / (2 * h)
}

# Punto final: backward
roi[n] <- (conversion[n] - conversion[n - 1]) / h

# — 2) Rango donde ROI > 0.2 (% por $k) —
umbral <- 0.2
indices_mayor_umbral <- which(roi > umbral)
puntos_cumplen <- gasto_k[indices_mayor_umbral]

# — 3) Segunda derivada en gasto = 15k (indice correspondiente) —
# indice de 15k es 4 (en R, 1-based)
i15 <- which(gasto_k == 15)
f2_15 <- (conversion[i15 + 1] - 2 * conversion[i15] + conversion[i15 - 1])

# — 4) Recomendacion simple basada en valores —
roi_25 <- roi[n]

# — Mostrar resultados —
cat("=====\n")
cat("EJERCICIO 6 – ROI marginal (R)\n")
cat("Datos (gasto en $k, conversion en %)\n")
cat("-----\n")
cat("(Tarea 1) ROI Marginal Calculado:\n")
for (i in 1:n) {
  cat(sprintf(" Gasto %2dk : Conversion = %.2f%% ROI approx %.4g %%/ $k\n"))
}
cat("-----\n")
cat(sprintf("(Tarea 2) Puntos donde ROI > %.2g %%/ $k : %s (en $k)\n", umbral), "\n")
cat(sprintf("(Tarea 3) Segunda derivada en $15k: f''(15) approx %.6g %%/ $k\n", f2_15), "\n")
cat(sprintf("(Tarea 4) ROI marginal en $25k: %.4g %%/ $k\n\n", roi_25))

cat("Interpretacion y recomendacion (Tarea 4):\n")
cat("- f''(15) negativo (", sprintf("%.4g", f2_15), ") indica rendimiento negativo\n")
cat("- Recomendacion: no aumentar gasto mas alla de $25k (ROI marginal negativo)\n")
cat("=====\n")
```



### 6.3. Salida de la Consola

La ejecución del script en R produce la siguiente salida.

Listing 12: Salida de la consola de R para el Ejercicio 6.

```

=====
EJERCICIO 6 – ROI marginal (R)
Datos (gasto en $k, conversion en %)
-----
(Tarea 1) ROI Marginal Calculado:
Gasto 0k : Conversion = 2.10%   ROI approx 0.34 %/ $k
Gasto 5k : Conversion = 3.80%   ROI approx 0.31 %/ $k
Gasto 10k : Conversion = 5.20%  ROI approx 0.23 %/ $k
Gasto 15k : Conversion = 6.10%  ROI approx 0.15 %/ $k
Gasto 20k : Conversion = 6.70%  ROI approx 0.09 %/ $k
Gasto 25k : Conversion = 7.00%  ROI approx 0.06 %/ $k
-----
(Tarea 2) Puntos donde ROI > 0.2 %/ $k : 0, 5, 10 (en $k)
(Tarea 3) Segunda derivada en $15k: f''(15) approx -0.012 % / ($k)^2
(Tarea 4) ROI marginal en $25k: 0.06 %/ $k

Interpretacion y recomendacion (Tarea 4):
- f''(15) negativo (-0.012) indica rendimientos decrecientes.
- Recomendacion: no aumentar gasto mas alla de $25k (ROI marginal muy baj
=====

```

## 7. Ejercicio 7: Feature Engineering con Derivadas

Tienes un dataset con la señal de un sensor de temperatura en un proceso industrial medida cada segundo:

Cuadro 8: Temperatura de sensor por segundo

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

#### Tareas:

1. Crea una nueva feature: la velocidad de cambio de temperatura (primera derivada) en cada punto.
2. Crea otra feature: la aceleración del cambio (segunda derivada).
3. Un aumento de temperatura mayor a 0,8 °C/s indica un problema. ¿En qué momentos se detectaría esta alerta?
4. Normaliza las features derivadas usando min-max scaling.



5. Explica por qué estas features derivadas pueden ser útiles para un modelo de clasificación que detecta anomalías.

## 7.1. Solución del Ejercicio 7

### Datos Iniciales del Problema

Se definen los datos como  $T(t)$  donde  $t$  es el tiempo en segundos.

- $T_0 = 20,1, T_1 = 20,3, T_2 = 20,8, T_3 = 21,5, T_4 = 22,6, T_5 = 24,2, T_6 = 26,1, T_7 = 28,5.$
- El paso (distancia entre puntos) es:  $h = 1$  segundo.

Las unidades serán  $^{\circ}\text{C/s}$  para la primera derivada y  $^{\circ}\text{C/s}^2$  para la segunda.

#### 7.1.1. Tarea 1: Feature 1 - Velocidad de Cambio (1ª Derivada)

Se usa diferencia hacia adelante en  $t = 0$ , hacia atrás en  $t = 7$ , y centrada en los puntos internos.

- **t=0 (Adelante):**  $f'(0) \approx \frac{T_1 - T_0}{h} = \frac{20,3 - 20,1}{1} = 0,20 \text{ }^{\circ}\text{C/s}$
- **t=1 (Centrada):**  $f'(1) \approx \frac{T_2 - T_0}{2h} = \frac{20,8 - 20,1}{2} = 0,35 \text{ }^{\circ}\text{C/s}$
- **t=2 (Centrada):**  $f'(2) \approx \frac{T_3 - T_1}{2h} = \frac{21,5 - 20,3}{2} = 0,60 \text{ }^{\circ}\text{C/s}$
- **t=3 (Centrada):**  $f'(3) \approx \frac{T_4 - T_2}{2h} = \frac{22,6 - 20,8}{2} = 0,90 \text{ }^{\circ}\text{C/s}$
- **t=4 (Centrada):**  $f'(4) \approx \frac{T_5 - T_3}{2h} = \frac{24,2 - 21,5}{2} = 1,35 \text{ }^{\circ}\text{C/s}$
- **t=5 (Centrada):**  $f'(5) \approx \frac{T_6 - T_4}{2h} = \frac{26,1 - 22,6}{2} = 1,75 \text{ }^{\circ}\text{C/s}$
- **t=6 (Centrada):**  $f'(6) \approx \frac{T_7 - T_5}{2h} = \frac{28,5 - 24,2}{2} = 2,15 \text{ }^{\circ}\text{C/s}$
- **t=7 (Atrás):**  $f'(7) \approx \frac{T_7 - T_6}{h} = \frac{28,5 - 26,1}{1} = 2,40 \text{ }^{\circ}\text{C/s}$

**Feature  $f'(t)$ :** [0.20, 0.35, 0.60, 0.90, 1.35, 1.75, 2.15, 2.40]

#### 7.1.2. Tarea 2: Feature 2 - Aceleración (2ª Derivada)

Se calcula para los puntos internos ( $t = 1$  a  $t = 6$ ) usando  $f''(t_i) \approx T_{i+1} - 2T_i + T_{i-1}$  (ya que  $h = 1$ ).

- **t=1:**  $f''(1) = T_2 - 2T_1 + T_0 = 20,8 - 2(20,3) + 20,1 = 0,30 \text{ }^{\circ}\text{C/s}^2$
- **t=2:**  $f''(2) = T_3 - 2T_2 + T_1 = 21,5 - 2(20,8) + 20,3 = 0,20 \text{ }^{\circ}\text{C/s}^2$
- **t=3:**  $f''(3) = T_4 - 2T_3 + T_2 = 22,6 - 2(21,5) + 20,8 = 0,40 \text{ }^{\circ}\text{C/s}^2$
- **t=4:**  $f''(4) = T_5 - 2T_4 + T_3 = 24,2 - 2(22,6) + 21,5 = 0,50 \text{ }^{\circ}\text{C/s}^2$
- **t=5:**  $f''(5) = T_6 - 2T_5 + T_4 = 26,1 - 2(24,2) + 22,6 = 0,30 \text{ }^{\circ}\text{C/s}^2$
- **t=6:**  $f''(6) = T_7 - 2T_6 + T_5 = 28,5 - 2(26,1) + 24,2 = 0,50 \text{ }^{\circ}\text{C/s}^2$

**Feature  $f''(t)$ :** [NA, 0.30, 0.20, 0.40, 0.50, 0.30, 0.50, NA]



### 7.1.3. Tarea 3: Detección de Alertas ( $f'(t) > 0,8 \text{ °C/s}$ )

Se comparan los valores de la Tarea 1 con el umbral de  $0.8 \text{ °C/s}$ .

- $t = 3 : 0,90 > 0,8$  (**Alerta**)
- $t = 4 : 1,35 > 0,8$  (**Alerta**)
- $t = 5 : 1,75 > 0,8$  (**Alerta**)
- $t = 6 : 2,15 > 0,8$  (**Alerta**)
- $t = 7 : 2,40 > 0,8$  (**Alerta**)

**Resultado:** Se detectarían alertas en los segundos **3, 4, 5, 6 y 7**.

### 7.1.4. Tarea 4: Normalización Min-Max

Se aplica la fórmula  $\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$  a cada feature.

**Para  $f'(t)$  (Velocidad):**  $\min = 0,20$ ,  $\max = 2,40$ . Rango = 2.20.

- $\tilde{f}'(t) \approx [0.000, 0.068, 0.182, 0.318, 0.523, 0.705, 0.886, 1.000]$

**Para  $f''(t)$  (Aceleración, en puntos 1-6):**  $\min = 0,20$ ,  $\max = 0,50$ . Rango = 0.30.

- $\tilde{f}''(t) \approx [\text{NA}, 0.333, 0.000, 0.667, 1.000, 0.333, 1.000, \text{NA}]$

### 7.1.5. Tarea 5: Utilidad para Modelos de Anomalías

Estas features (características) derivadas son fundamentales para un modelo de clasificación de anomalías:

- **Velocidad (1ª Derivada):** Un modelo que solo mira la temperatura (ej.  $T = 21,5$ ) no ve problemas. Pero la *velocidad* (ej.  $f'(3) = 0,9$ ) sí. La velocidad detecta **cambios súbitos** (como el inicio de un sobrecalentamiento) que la temperatura absoluta aún no refleja.
- **Aceleración (2ª Derivada):** La aceleración mide si el problema está **empeorando o estabilizándose**. Una  $f'' > 0$  (como  $f''(4) = 0,5$ ) indica que la velocidad de calentamiento está, a su vez, aumentando. Esto permite al modelo anticipar tendencias peligrosas.
- **Normalización:** Poner ambas features en la misma escala  $[0, 1]$  es crucial para que el modelo (como SVM, Redes Neuronales o k-NN) no le dé más peso a una feature solo por tener números más grandes.

## 7.2. Código en R para Verificación

Script en R que implementa y verifica los cálculos, generando un `data.frame` con todas las features.



Listing 13: Código en R para el Ejercicio 7 (Feature Engineering).

```
# =====  
# EJERCICIO 7: Feature Engineering con Derivadas (Sensor)  
# Programacion Numerica – Diferenciacion Numerica  
# =====  
  
# — Datos del PDF —  
t <- 0:7  
temp <- c(20.1, 20.3, 20.8, 21.5, 22.6, 24.2, 26.1, 28.5) # en C  
h <- 1 # paso en segundos  
  
n <- length(t)  
  
# — 1) Primera derivada (velocidad, C /s) —  
d1 <- rep(NA_real_, n)  
# Adelante en el primero  
d1[1] <- (temp[2] - temp[1]) / h  
# Centrada en internos  
for (i in 2:(n-1)) {  
  d1[i] <- (temp[i + 1] - temp[i - 1]) / (2 * h)  
}  
# Atras en el ultimo  
d1[n] <- (temp[n] - temp[n - 1]) / h  
  
# — 2) Segunda derivada (aceleracion, C /s^2) en internos —  
d2 <- rep(NA_real_, n)  
for (i in 2:(n-1)) {  
  d2[i] <- (temp[i + 1] - 2 * temp[i] + temp[i - 1]) / (h^2)  
}  
  
# — 3) Alertas por umbral —  
umbral <- 0.8 # C /s  
alertas_idx <- which(d1 > umbral)  
alertas_t <- t[alertas_idx]  
  
# — 4) Normalizacion min-max por feature —  
minmax <- function(x) {  
  # ignora NAs para calcular min/max  
  xr <- range(x, na.rm = TRUE)  
  if (xr[2] == xr[1]) return((x - xr[1]) * 0) # evita division por 0  
  (x - xr[1]) / (xr[2] - xr[1])  
}  
d1_minmax <- minmax(d1)  
# Se aplica minmax solo a los valores no-NA de d2  
d2_minmax_calc <- minmax(d2[!is.na(d2)])  
# Se re-insertan los NAs en las posiciones originales  
d2_minmax <- rep(NA_real_, n)  
d2_minmax[!is.na(d2)] <- d2_minmax_calc
```



```
# — Resultados en una tabla —
```

```
resultado <- data.frame(  
  tiempo_s = t,  
  temp_C = temp,  
  d1_C_por_s = round(d1, 3),  
  d2_C_por_s2 = round(d2, 3),  
  d1_minmax = round(d1_minmax, 3),  
  d2_minmax = round(d2_minmax, 3),  
  alerta_d1_gt_0_8 = (d1 > umbral)  
)
```

```
print(resultado, row.names = FALSE)
```

```
cat("\nUmbral de alerta (d1 > ", umbral, " C /s) en t = {", paste(alerta
```

### 7.3. Salida de la Consola

La ejecución del script en R produce el siguiente data.frame y resumen de alertas.

Listing 14: Salida de la consola de R para el Ejercicio 7.

	tiempo_s	temp_C	d1_C_por_s	d2_C_por_s2	d1_minmax	d2_minmax	alerta_d1_gt_0_8
	0	20.1	0.200	NA	0.000	NA	
FALSE	1	20.3	0.350	0.300	0.068	0.333	
FALSE	2	20.8	0.600	0.200	0.182	0.000	
FALSE	3	21.5	0.900	0.400	0.318	0.667	
TRUE	4	22.6	1.350	0.500	0.523	1.000	
TRUE	5	24.2	1.750	0.300	0.705	0.333	
TRUE	6	26.1	2.150	0.500	0.886	1.000	
TRUE	7	28.5	2.400	NA	1.000	NA	

```
Umbral de alerta (d1 > 0.8 C /s) en t = {3, 4, 5, 6, 7} s
```