Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente: Ing. Fred Torres Cruz
Estudiante: Luis Angel Quenaya Loza
Código: 241411

Trabajo N°11: Ejercicios Aplicados a Data Science

Curso: Programación Numérica

1. Fundamentos de la Diferenciación Numérica

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

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

En práctica usamos diferencias finitas:

Diferencia hacia adelante (forward):

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

Diferencia hacia atrás (backward):

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

Diferencia centrada (central):

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

Segunda derivada:

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

Estos son los métodos que aplicaremos, eligiendo h apropiado según el problema (por ejemplo $h = 1$ para datos discretos por unidad, o un h pequeño para funciones continuas).

2. Ejercicio 1: Diferencias Finitas e Incremento de Usuarios

(Enunciado tal cual lo recibiste.)

Resolución (pasos necesarios usando las fórmulas):

Datos:

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

Queremos la tasa en el mes 4 usando diferencia centrada con $h = 1$:

$$f'(4) \approx \frac{f(5) - f(3)}{2 \cdot 1} = \frac{48 - 23}{2} = 12{,}5 \text{ (k/mes)}.$$

Adelante en mes 1 ($h = 1$):

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

Atrás en mes 7 ($h = 1$):

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

Segunda derivada (aceleración) con $h = 1$:

$$f''(i) \approx f(i+1) - 2f(i) + f(i-1).$$

```r
library(ggplot2)
mes <- c(1,2,3,4,5,6,7)
usuarios <- c(10,15,23,34,48,65,85)
tasa_centrada_mes4 <- (usuarios[5]-usuarios[3])/(2*1)
tasa_adelante_mes1 <- (usuarios[2]-usuarios[1])/1
tasa_atras_mes7 <- (usuarios[7]-usuarios[6])/1
segunda <- c(NA, diff(usuarios, differences = 2), NA)
datos <- data.frame(mes, usuarios)
ggplot(datos, aes(x = mes, y = usuarios)) +
  geom_line(color = "blue") +
  geom_point(color = "red", size = 3) +
  ggtitle("Crecimiento de Usuarios") +
  xlab("Mes") + ylab("Usuarios (miles)")
tasa_centrada_mes4; tasa_adelante_mes1; tasa_atras_mes7; segunda
```

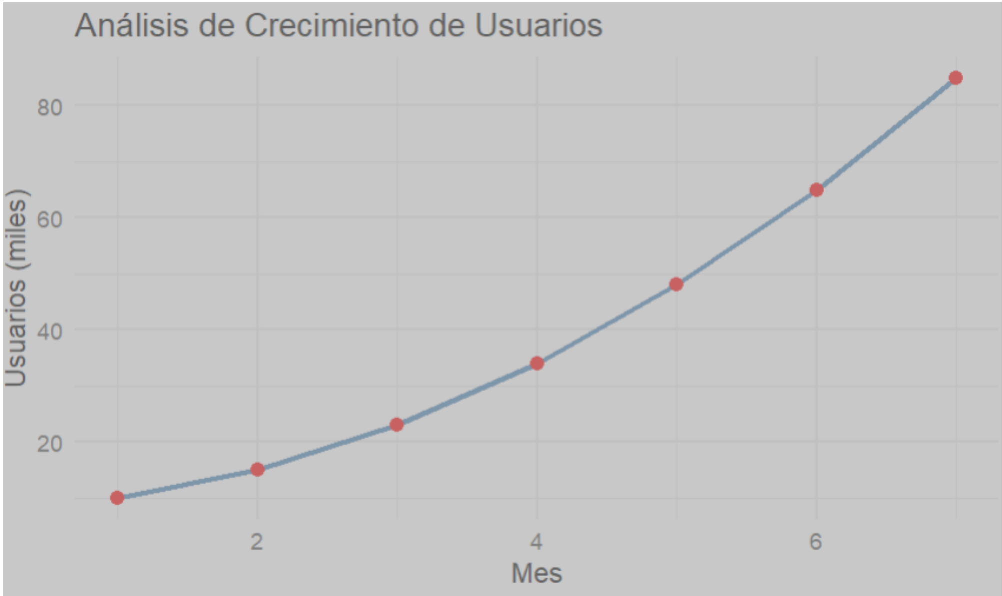

Figura 1: Gráfico de crecimiento de usuarios.

3. Ejercicio 2: Optimización de Función de Pérdida

(Enunciado tal cual lo recibiste.)
Resolución (pasos necesarios usando las fórmulas):
Datos:

Época	0	10	20	30	40	50
Loss	2,45	1,82	1,35	1,08	0,95	0,89

Tasa centrada en $x = 20$ con $h = 10$:

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

Segunda derivada en $x = 30$ con $h = 10$:

$$f''(30) \approx \frac{f(40) - 2f(30) + f(20)}{10^2} = \frac{0{,}95 - 2(1{,}08) + 1{,}35}{100} = 0{,}0014.$$

Criterio $|f'(x)| < 0{,}01$: calculamos derivadas por diferencias finitas y buscamos la primera época que cumple.

Interpolación lineal para $x = 25$ (entre 20 y 30):

$$f(25) = f(20) + \frac{f(30) - f(20)}{30 - 20} \cdot (25 - 20) = 1{,}35 + \frac{1{,}08 - 1{,}35}{10} \cdot 5 = 1{,}215.$$

```r
library(ggplot2)
epoca <- c(0,10,20,30,40,50)
loss <- c(2.45,1.82,1.35,1.08,0.95,0.89)
h <- 10
```

```
primera <- numeric(length(loss))
segunda <- numeric(length(loss))
for (i in 2:(length(loss)-1)) {
  primera[i] <- (loss[i+1] - loss[i-1]) / (2*h)
  segunda[i] <- (loss[i+1] - 2*loss[i] + loss[i-1]) / (h^2)
}
epoca_detener <- epoca[which(abs(primera) < 0.01)[1]]
loss_25 <- loss[3] + (loss[4]-loss[3])*(25-20)/(30-20)
datos <- data.frame(epoca, loss)
ggplot(datos, aes(x = epoca, y = loss)) +
  geom_line(color = "blue") +
  geom_point(size = 3, color = "red") +
  ggtitle("Evoluci n del loss") +
  xlab(" poca ") + ylab("Loss")
primera; segunda; epoca_detener; loss_25
```

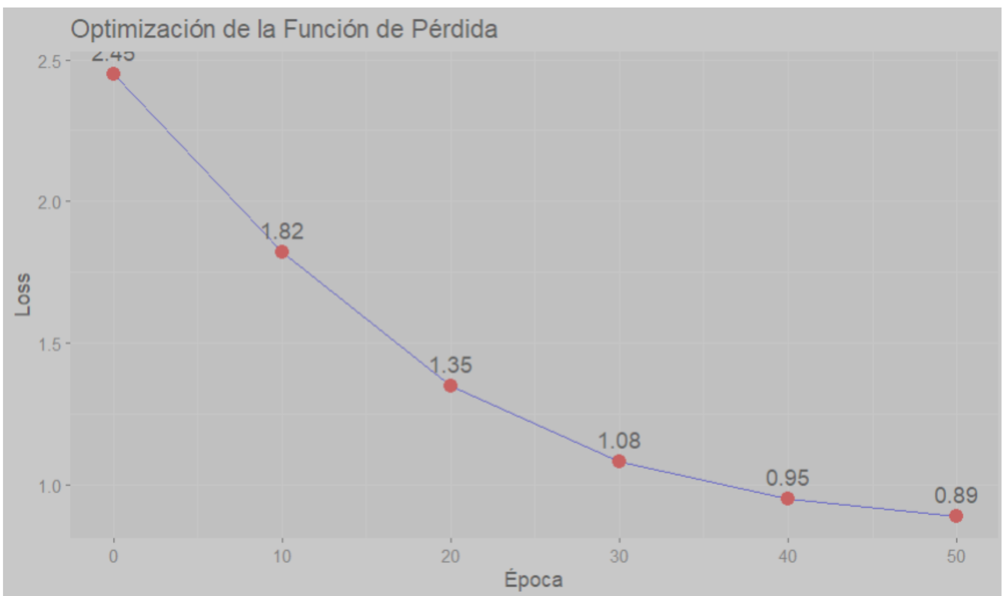

Figura 2: Evolución del loss durante el entrenamiento.

4. Ejercicio 3: Análisis de Series Temporales de Ventas

(Enunciado tal cual lo recibiste.)

Resolución (pasos necesarios usando las fórmulas):

Datos:

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

Primera derivada (velocidad) con $h = 1$ (centrada donde sea posible):

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

Segunda derivada:

$$f''(i) \approx f(i+1) - 2f(i) + f(i-1).$$

Desaceleración del jueves: $f''(Jue) = f(Vie) - 2f(Jue) + f(Mié)$. Extrapolación para lunes siguiente: $f(Lun\ siguiente) = f(Dom) + f'(Dom)$.

```
library(ggplot2)
dia <- c("Lun","Mar","Mi ","Jue","Vie","S b","Dom")
ventas <- c(45,52,61,58,73,89,95)
n <- length(ventas)
primera <- numeric(n)
segunda <- numeric(n)
for (i in 1:n) {
  if (i == 1) {
    primera[i] <- ventas[i+1]-ventas[i]
  } else if (i == n) {
    primera[i] <- ventas[i]-ventas[i-1]
  } else {
    primera[i] <- (ventas[i+1]-ventas[i-1]) / 2
  }
}
for (i in 2:(n-1)) {
  segunda[i] <- ventas[i+1] - 2*ventas[i] + ventas[i-1]
}
dia_max_acel <- dia[which.max(segunda)]
desac_jue <- segunda[4]
pred_lunes <- ventas[7] + primera[7]
df <- data.frame(dia, ventas, primera, segunda)
ggplot(df, aes(x = factor(dia, levels=dia), y = ventas, group=1)) +
  geom_line(color="blue") + geom_point(color="red", size=3) +
  ggtitle("Ventas diarias durante la campa a") + xlab("D a") +
     ylab("Ventas ($k)")
dia_max_acel; desac_jue; pred_lunes
```

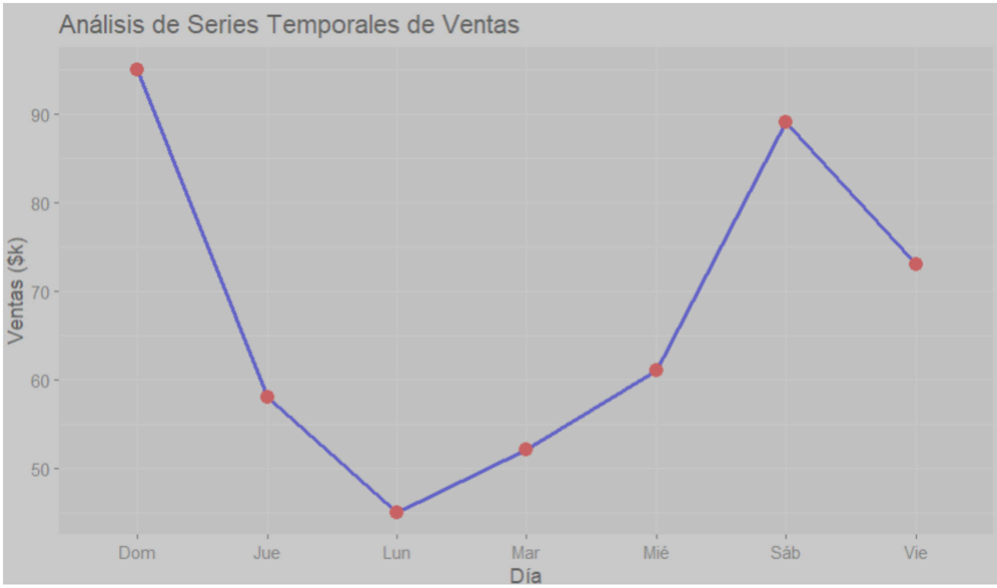

Figura 3: Evolución de las ventas durante la semana de campaña.

5. Ejercicio 4: Gradiente de Función de Activación

(Enunciado tal cual lo recibiste.)
Resolución (pasos necesarios usando las fórmulas):
Datos:

$$x = [-3, -2, -1, 0, 1, 2, 3], \quad \sigma(x) = [0{,}0474, 0{,}1192, 0{,}2689, 0{,}5, 0{,}7311, 0{,}8808, 0{,}9526].$$

Usar diferencia centrada ($h = 1$):

$$\sigma'(x_i) \approx \frac{\sigma(x_{i+1}) - \sigma(x_{i-1})}{2h}.$$

Comparar con la derivada analítica:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)).$$

```
library(ggplot2)
x <- c(-3,-2,-1,0,1,2,3)
sigma <- c(0.0474,0.1192,0.2689,0.5,0.7311,0.8808,0.9526)
h <- 1
sigma_p <- numeric(length(x))
for (i in 1:length(x)) {
  if (i == 1) {
    sigma_p[i] <- (sigma[i+1]-sigma[i]) / h
  } else if (i == length(x)) {
    sigma_p[i] <- (sigma[i]-sigma[i-1]) / h
  } else {
```

```
    sigma_p[i] <- (sigma[i+1]-sigma[i-1]) / (2*h)
  }
}
sigma_analitica <- sigma*(1-sigma)
df <- data.frame(x, sigma, sigma_p, sigma_analitica)
ggplot(df, aes(x = x)) +
  geom_line(aes(y = sigma_p), color = "blue", size = 1.2) +
  geom_point(aes(y = sigma_p), color = "red", size = 3) +
  geom_line(aes(y = sigma_analitica), color = "orange", linetype="
     dashed", size=1) +
  ggtitle("Gradiente de la funci n sigmoide") + xlab("x") + ylab("
       '(x)")
df
```

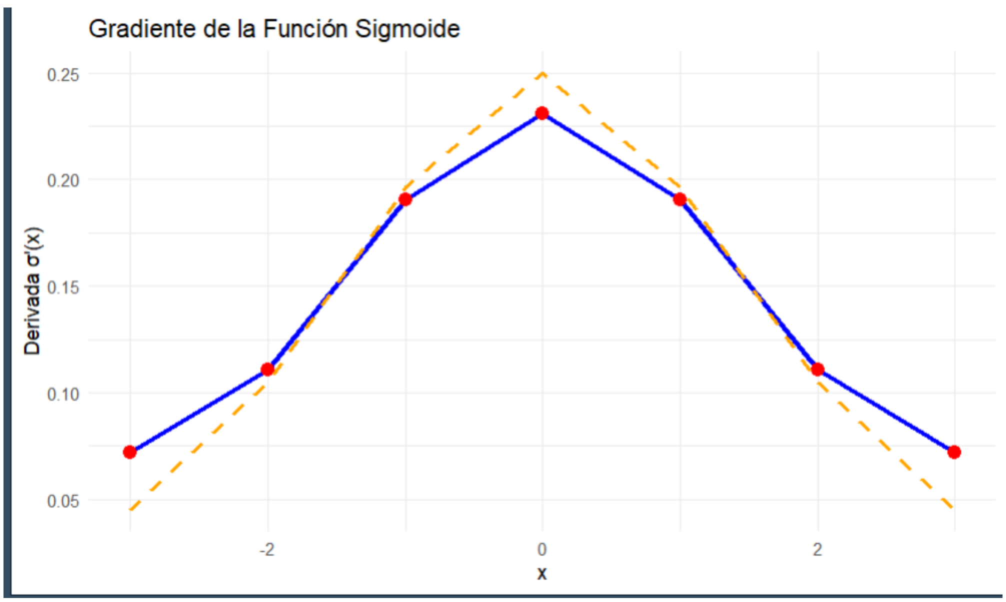

Figura 4: Gradiente y derivada analítica de la sigmoide.

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

(Enunciado tal cual lo recibiste.)
Resolución (pasos necesarios usando las fórmulas):
Datos:

$$t = 0, \ldots, 7, \quad L = [120, 125, 128, 135, 280, 290, 275, 155].$$

Primera derivada (centrada donde aplique):

$$L'(i) \approx \frac{L_{i+1} - L_{i-1}}{2}.$$

Segunda derivada:

$$L''(i) \approx L_{i+1} - 2L_i + L_{i-1}.$$

Salto entre 3 y 4: $L_4 - L_3$. Anomalía si $|L'| > 50$.

```r
library(ggplot2)
hora <- c(0,1,2,3,4,5,6,7)
latencia <- c(120,125,128,135,280,290,275,155)
n <- length(latencia)
primera <- numeric(n)
segunda <- numeric(n)
for (i in 1:n) {
  if (i == 1) {
    primera[i] <- latencia[i+1]-latencia[i]
  } else if (i == n) {
    primera[i] <- latencia[i]-latencia[i-1]
  } else {
    primera[i] <- (latencia[i+1]-latencia[i-1]) / 2
  }
}
for (i in 2:(n-1)) {
  segunda[i] <- latencia[i+1] - 2*latencia[i] + latencia[i-1]
}
salto_34 <- latencia[5] - latencia[4]
tasa_recup <- (latencia[8] - latencia[7])/(hora[8]-hora[7])
anomalias <- which(abs(primera) > 50)
df5 <- data.frame(hora, latencia, primera, segunda)
ggplot(df5, aes(x=hora, y=latencia)) +
  geom_line(color="blue") + geom_point(color="red", size=3) +
  geom_point(data = df5[anomalias, ], aes(x=hora, y=latencia), color="orange", size=4) +
  ggtitle("Monitoreo de latencia del sistema") + xlab("Hora") + ylab("Latencia (ms)")
salto_34; tasa_recup; anomalias
```

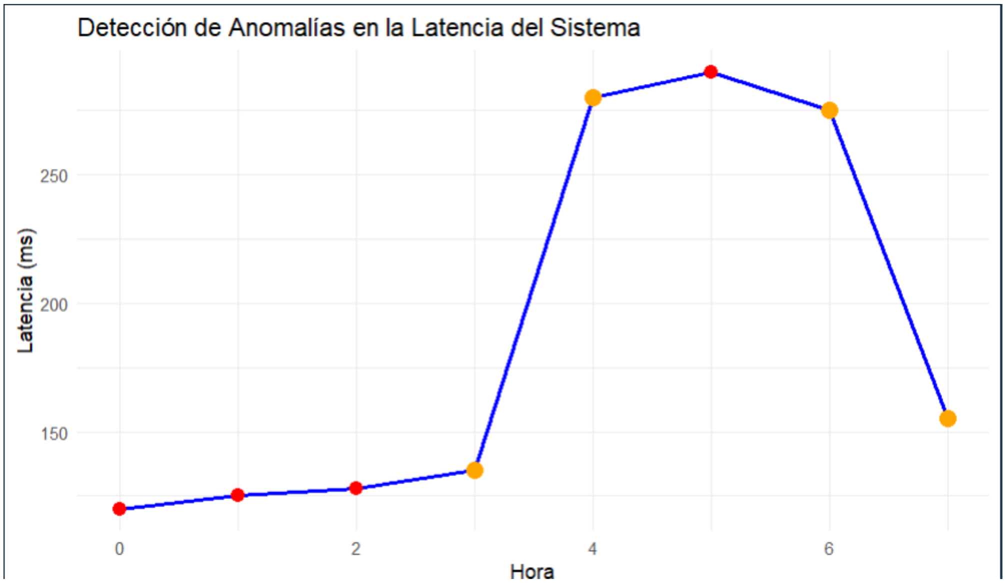

Figura 5: Detección de anomalías en latencia.

7. Ejercicio 6: Análisis de Tasa de Conversión (ROI marginal)

(Enunciado tal cual lo recibiste.)
Resolución (pasos necesarios usando las fórmulas):
Datos:

$$G = [0, 5, 10, 15, 20, 25], \quad C = [2{,}1, 3{,}8, 5{,}2, 6{,}1, 6{,}7, 7{,}0].$$

ROI marginal (diferencia centrada, $h = 5$):

$$ROI'(G_i) \approx \frac{C(G_{i+1}) - C(G_{i-1})}{2h}.$$

Segunda derivada:

$$ROI''(G_i) \approx \frac{C(G_{i+1}) - 2C(G_i) + C(G_{i-1})}{h^2}.$$

Buscar $ROI' > 0{,}2$ (por cada \$1k).

```
library(ggplot2)
gasto <- c(0,5,10,15,20,25)
conversion <- c(2.1,3.8,5.2,6.1,6.7,7.0)
n <- length(gasto)
roi_marginal <- numeric(n)
roi_segunda <- numeric(n)
for (i in 1:n) {
  if (i == 1) {
    roi_marginal[i] <- (conversion[i+1]-conversion[i]) / (gasto[i
       +1]-gasto[i])
```

```r
  } else if (i == n) {
    roi_marginal[i] <- (conversion[i]-conversion[i-1]) / (gasto[i]-
       gasto[i-1])
  } else {
    roi_marginal[i] <- (conversion[i+1]-conversion[i-1]) / (2*(gasto
       [i+1]-gasto[i]))
  }
}
for (i in 2:(n-1)) {
  roi_segunda[i] <- (conversion[i+1]-2*conversion[i]+conversion[i
     -1]) / ((gasto[2]-gasto[1])^2)
}
rango_mayor_02 <- gasto[which(roi_marginal > 0.2)]
df6 <- data.frame(gasto, conversion, roi_marginal, roi_segunda)
ggplot(df6, aes(x=gasto, y=conversion)) +
  geom_line(color="blue") + geom_point(color="red", size=3) +
  ggtitle("Tasa de conversi n vs gasto") + xlab("Gasto ($k)") +
     ylab("Conversi n (%)")
roi_marginal; roi_segunda[which(gasto==15)]; rango_mayor_02
```

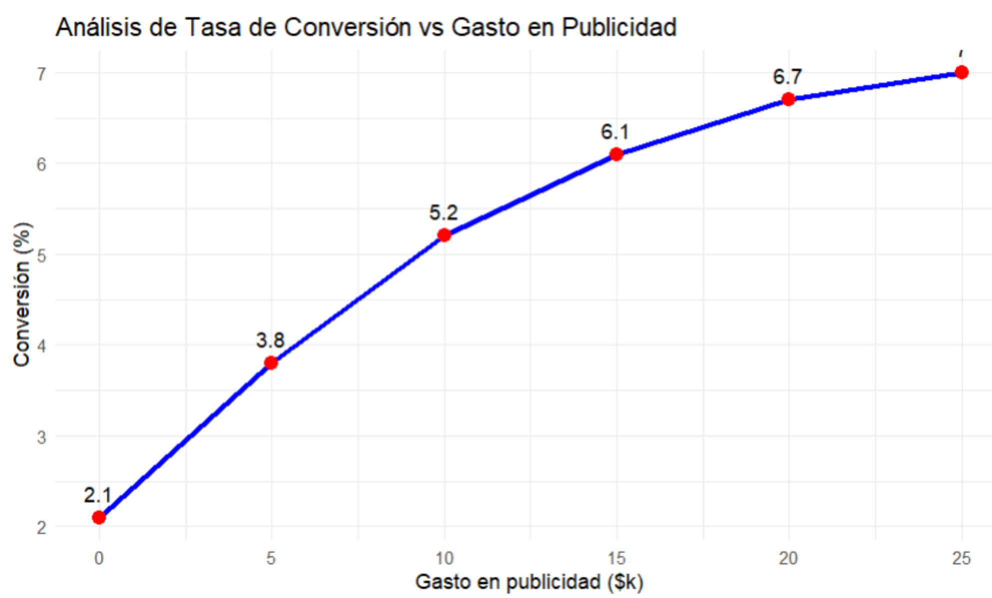

Figura 6: ROI marginal y conversión.

8. Ejercicio 7: Feature Engineering con Derivadas (Sensor de Temperatura)

(Enunciado tal cual lo recibiste.)

Resolución (pasos necesarios usando las fórmulas):

Datos:

$$t = [0, \ldots, 7], \quad T = [20{,}1, 20{,}3, 20{,}8, 21{,}5, 22{,}6, 24{,}2, 26{,}1, 28{,}5].$$

Velocidad (primera derivada centrada donde se pueda):

$$v_i \approx \frac{T_{i+1} - T_{i-1}}{2}.$$

Aceleración (segunda derivada):

$$a_i \approx T_{i+1} - 2T_i + T_{i-1}.$$

Alerta si $v_i > 0{,}8$. Normalización min-max.

```r
library(ggplot2)
tiempo <- c(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)
n <- length(temp)
velocidad <- numeric(n)
for (i in 1:n) {
  if (i == 1) {
    velocidad[i] <- temp[i+1]-temp[i]
  } else if (i == n) {
    velocidad[i] <- temp[i]-temp[i-1]
  } else {
    velocidad[i] <- (temp[i+1]-temp[i-1]) / 2
  }
}
aceleracion <- numeric(n)
for (i in 2:(n-1)) {
  aceleracion[i] <- temp[i+1] - 2*temp[i] + temp[i-1]
}
alertas <- which(velocidad > 0.8)
vel_norm <- (velocidad - min(velocidad)) / (max(velocidad) - min(
   velocidad))
acel_norm <- (aceleracion - min(aceleracion, na.rm=TRUE)) / (max(
   aceleracion, na.rm=TRUE) - min(aceleracion, na.rm=TRUE))
df7 <- data.frame(tiempo, temp, velocidad, aceleracion, vel_norm,
   acel_norm)
ggplot(df7, aes(x=tiempo, y=temp)) +
  geom_line(color="blue") + geom_point(color="red", size=3) +
  ggtitle("Sensor de temperatura y derivadas") + xlab("Tiempo (s)")
     + ylab("Temp ( C )")
velocidad; aceleracion; alertas
```

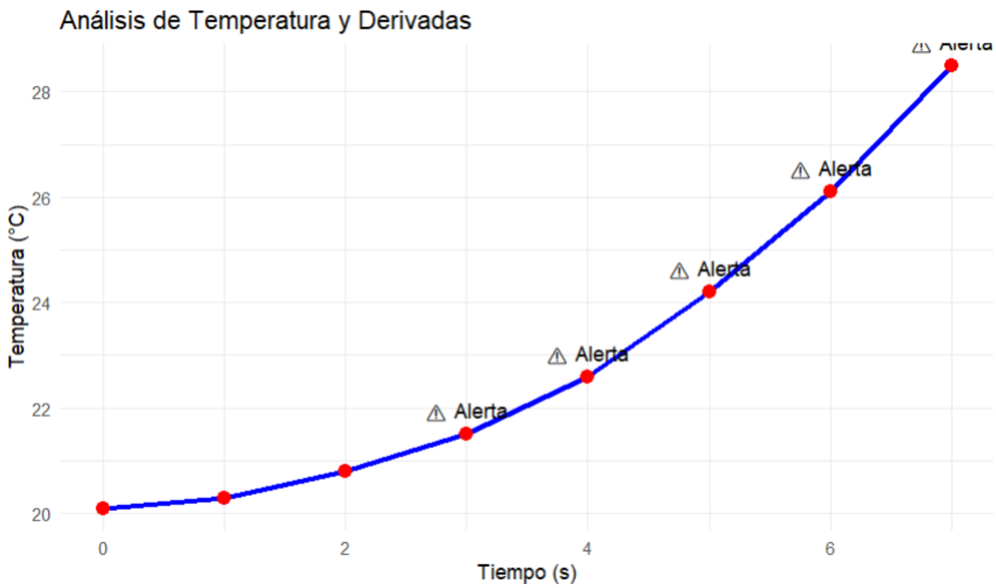

Figura 7: Velocidad y aceleración de la señal de temperatura.