

EJERCICIOS R NIVEL BÁSICO

En el siguiente recurso se revisarán los códigos vistos en clase correspondientes a un nivel **básico e intermedio**:

- Operaciones aritméticas (suma, resta, producto, división y potencia)
- Funciones matemáticas (exponencial, logaritmo, trigonométricas, valor absoluto, raíz cuadrada)
- Codificación
- Condicionales (bucle for, bucle if, bucle while)

En primer lugar, se **enunciarán** las sentencias y elementos de código correspondientes a cada categoría¹. A continuación, se propondrá un **ejercicio** en el que haya que poner en práctica lo aprendido. Este ejercicio se encontrará dentro del marco de la investigación en biotecnología con el fin de ilustrar la importancia que tiene R como herramienta de tratamiento de datos en el contexto de la investigación.

OPERACIONES ARITMÉTICAS

Sentencias:

- Suma: +
- Resta: -
- Producto: *
- División: /

Enunciado:

En un experimento realizado por alumnos universitarios se observa la actividad enzimática de la lactasa en unas condiciones de temperatura, pH del medio, concentración de sustrato y de enzima específicas. Se desea comparar los datos obtenidos de actividad enzimática con valores de experimentos realizados por profesionales en las mismas condiciones de temperatura, pH y concentración. A continuación, se muestran los resultados obtenidos:

Valor aproximado = 0,014 $\mu\text{mol}/\text{min}$

Valor real = 0,013 $\mu\text{mol}/\text{min}$

Para determinar la validez de los resultados, emplea el cálculo del error relativo.

¹ Existen ciertas sentencias comunes para casi todos los códigos que se explicarán únicamente la primera vez que aparezcan.

Solución:

```
1 # Definir los valores obtenidos y los valores reales
2 valor_aproximado <- 0.014
3 valor_real <- 0.013
4
5 # Calcular el error absoluto
6 error_absoluto <- abs(valor_aproximado - valor_real)
7
8 # Calcular el error relativo (en porcentaje)
9 error_relativo <- (error_absoluto / valor_real) * 100
10
11 # Mostrar el resultado
12 cat("El error relativo es:", error_relativo, "%\n")
```

FUNCIONES MATEMÁTICAS

Sentencias:

- Exponencial: $\exp(x)$
- Logaritmo: $\log(x)$
- Seno: $\sin(x)$
- Coseno: $\cos(x)$
- Tangente: $\tan(x)$
- Valor absoluto: $\text{abs}(x)$
- Raíz cuadrada: $\text{sqrt}(x)$
- Gráfico de dispersión: $\text{plot}(x,y)$

Enunciado 1:

El nivel de melatonina en el organismo, que regula nuestro ritmo circadiano, varía de manera cíclica. Se ha encontrado la siguiente función sinusoidal que aproxima la concentración plasmática de melatonina en un adulto medio en función de la hora del día:

$$C(t) = A \sin\left(\frac{2\pi}{24}(t - \varphi)\right) + C_m$$

Donde:

- $C(t)$ es la concentración de melatonina en función de la hora del día t .
- $A = 50$ g/mol
- $\varphi = 2$ horas
- $C_m = 30$ g/mol

Calcular la concentración de melatonina en el organismo a las 12:00.

Solución 1:

```
1 # Definir la función sinusoidal para la concentración de melatonina
2 concentracion_melatonina <- function(t) {
3   A <- 50
4   phi <- 2
5   Cm <- 30
6   C_t <- A * sin((2 * pi / 24) * (t - phi)) + Cm
7   return(C_t)
8 }
9
10 # Solicitar al usuario la hora del día
11 hora_dia <- as.numeric(readline(prompt = "Introduce la hora del día (0-23): "))
12
13 # Calcular la concentración de melatonina a la hora proporcionada
14 concentracion <- concentracion_melatonina(hora_dia)
15
16 # Mostrar el resultado
17 cat("La concentración de melatonina a las", hora_dia, "horas es aproximadamente", concentracion, "unidades.\n")
```

Enunciado 2:

En un experimento, se mide el crecimiento de una población bacteriana a lo largo del tiempo. La ecuación de crecimiento que modela el crecimiento es la siguiente:

$$N(t) = N_0 e^{\sqrt{kt}}$$

donde:

- $N(t)$ es el número de bacterias en el tiempo t .
- N_0 es el número inicial de bacterias.
- k es la constante de crecimiento.

Calcula y grafica el comportamiento del crecimiento bacteriano utilizando la ecuación dada para valores de t en el intervalo $[0, 20]$.

Solución 2:

```
1 # Paso 1: Definir los parámetros iniciales
2 N0 <- 100 # Número inicial de bacterias
3 k <- 0.1 # Constante de crecimiento
4
5 # Paso 2: Definir el rango de tiempo
6 t <- seq(0, 20, by = 0.1) # Tiempo de 0 a 20 con incrementos de 0.1
7
8 # Paso 3: Calcular el número de bacterias en función del tiempo
9 N_t <- N0 * exp(sqrt(k * t))
10
11 # Paso 4: Graficar resultados
12 plot(t, N_t, type = "l", col = "green", lwd = 2,
13       main = expression(N(t) == N[0] * e^{sqrt(kt)}),
14       xlab = "Tiempo (t)", ylab = "Número de bacterias (N(t))")
15
16 # Añadir una cuadrícula
17 grid()
```

CODIFICACIÓN

Sentencias:

- Quedarse con el valor entero de una división: floor

Enunciado:

En un ensayo clínico se desean almacenar los datos de cada paciente, incluyendo su número de teléfono. Por razones de confidencialidad se debe convertir el número a código binario para almacenarlo en la base de datos. Escribe un código que convierta el número a código binario.

Solución:

```
1 cat("\014")
2 rm(list=ls())
3 num<- as.numeric(readline(prompt = "Introduce el número de teléfono "))
4 #Inicializa un vector para almacenar los dígitos del número de teléfono
5 n_tel=c()
6 num_parcial=num
7 #Iniciar bucle que divide entre dos y se queda con el valor entero
8 while(num_parcial>0){
9   resto=num_parcial%%2
10  n_tel=c(resto,n_tel)
11  num_parcial=floor(num_parcial/2)
12 }
13 #Muestra en pantalla el número en código binario
14 print(n_tel)
```

CONDICIONALES

Sentencias:

- Bucle for ("para"): for
- Bucle if: if (si), else if (sino)
- Bucle while ("mientras"): while

Enunciado 1:

En un estudio de poblaciones genéticas, se desea determinar la frecuencia de un alelo específico ("A") en cada posición de la secuencia genética. En concreto, se estudian las siguientes secuencias de 5 individuos:

1. "ATCG"
2. "AAGC"
3. "ATAC"
4. "AAAG"
5. "ATCC"

Escribe un programa que procese las secuencias de ADN de varios individuos, determine la frecuencia del alelo "A" en cada posición de la secuencia y muestre los resultados.

Solución 1:

```
1 # Inicializar las secuencias de ADN de los individuos
2 secuencias_adn <- c("ATCG", "AAGC", "ATAC", "AAAG", "ATCC")
3
4 # Definir el alelo a buscar
5 alelo <- "A"
6
7 # Inicializar un vector para contar las frecuencias del alelo "A" en cada posición
8 frecuencias <- c(0, 0, 0, 0)
9
10 # Iterar sobre las posiciones de la secuencia de ADN usando un bucle for
11 for (posicion in 1:4) { # Longitud fija de las secuencias de ADN
12   contador <- 0
13
14   # Iterar sobre las secuencias de ADN de los individuos usando un bucle for anidado
15   for (secuencia in secuencias_adn) {
16     # Verificar si el alelo "A" está presente en la posición actual usando un bucle if
17     if (substr(secuencia, posicion, posicion) == alelo) {
18       contador <- contador + 1
19     }
20   }
21
22   # Calcular la frecuencia del alelo "A" en la posición actual
23   frecuencias[posicion] <- contador / length(secuencias_adn)
24 }
25
26 # Mostrar los resultados
27 for (posicion in 1:4) { # Longitud fija de las secuencias de ADN
28   cat("La frecuencia del alelo 'A' en la posición", posicion, "es", frecuencias[posicion], "\n")
29 }
```

Enunciado 2:

En un experimento de termodinámica, es importante que la temperatura ambiente del laboratorio se mantenga bajo los 35°C. Para monitorearla, se ha instalado un dispositivo que mide la temperatura cada minuto. La temperatura inicial es de 25°C y cada minuto aumenta entre 0.1°C y 0.5°C (de forma aleatoria). Escribe un programa que pare de monitorear la temperatura si se sube de 35°C y emita un aviso.

Solución 2:

```
1 # Definir la temperatura inicial
2 temperatura <- 25
3 # Contador de minutos
4 minutos <- 0
5
6 # Bucle para registrar la temperatura
7 while (temperatura <= 35) {
8   # Incrementar el tiempo en 1 minuto
9   minutos <- minutos + 1
10
11   # Aumentar la temperatura aleatoriamente entre 0.1°C y 0.5°C
12   incremento <- runif(1, min = 0.1, max = 0.5)
13   temperatura <- temperatura + incremento
14   print (temperatura)
15 }
16
17 # Imprimir el aviso
18 cat(sprintf("La temperatura superó los 35°C después de %d minutos.\n", minutos))
```