

Tarea #2: Variables, Funciones, Ciclos, Vectores y Matrices en R

Objetivo:

El objetivo de esta tarea es aplicar conceptos básicos en R como variables, funciones, operaciones, números complejos, ciclos, vectores y matrices. Trabajarás con sentencias condicionales, sentencias repetitivas, y funciones como `lapply()` y `sapply()`.

Parte 1: Variables y Tipos de Datos (15 puntos)

1. Declaración de Variables:
 - a. Declara las siguientes variables:
 - i. Un número entero llamado `entero`.
 - ii. Un número decimal llamado `decimal`.
 - iii. Un texto llamado `mensaje`.
 - iv. Un valor lógico llamado `es_verdadero`.
2. Operaciones con Variables:
 - a. Suma el número entero con el decimal.
 - b. Multiplica el entero por 5.
 - c. Imprime el valor lógico en un mensaje concatenado (usa `paste()`).

Parte 2: Funciones y Operaciones Básicas (15 puntos)

1. Definición de una Función Simple:
 - a. Crea una función llamada `calcular_doble` que acepte un número como parámetro y devuelva el doble de ese número.
2. Aplicación de la Función:
 - a. Utiliza la función `calcular_doble` para calcular el doble de entero y decimal.
3. Números Complejos:
 - a. Crea un número complejo `num_complejo` (por ejemplo, $3 + 2i$).
 - b. Multiplica este número por otro número complejo (por ejemplo, $2 + 3i$).

Parte 3: Sentencias Condicionales y Ciclos (20 puntos)

1. Condicionales con `if`, `else if`, `else`:
 - a. Escribe una sentencia condicional que compare el valor de entero con 10:
 - i. Si entero es mayor que 10, imprime 'Mayor a 10'.
 - ii. Si entero es igual a 10, imprime 'Es igual a 10'.
 - iii. Si entero es menor que 10, imprime 'Menor a 10'.

2. Ciclo for:
 - a. Crea un ciclo for que imprima los números del 1 al 5.
3. Ciclo while:
 - a. Utiliza un ciclo while que sume números consecutivos desde 1 hasta que la suma supere 15. Imprime la suma acumulada en cada iteración.

Parte 4: Uso de lapply() y sapply() (20 puntos)

1. Lista de Números:
 - a. Crea una lista llamada números que contenga los siguientes elementos: vector1 = c(1, 2, 3), vector2 = c(4, 5, 6), y vector3 = c(7, 8, 9).
2. Uso de lapply():
 - a. Utiliza lapply() para multiplicar cada número en la lista por 2.
3. Uso de sapply():
 - a. Utiliza sapply() para calcular la suma de los elementos de cada vector en la lista.

Parte 5: Vectores y Matrices (30 puntos)

1. Creación de un Vector y Operaciones:
 - a. Crea un vector mi_vector con los números del 1 al 10.
 - b. Realiza las siguientes operaciones:
 - i. Multiplica el vector por 3.
 - ii. Calcula el promedio (mean()) del vector.
2. Condicionales con Vectores:
 - a. Usa un ciclo for para imprimir si cada número del vector es par o impar.
3. Creación de una Matriz y Operaciones:
 - a. Crea una matriz mi_matriz de 3x3 con los números del 1 al 9.
 - b. Realiza las siguientes operaciones:
 - i. Multiplica la matriz por 2.
 - ii. Calcula la transposición de la matriz.

Parte 6: Análisis Estadístico de un Data Frame (10 puntos)

1. Creación de un Data Frame:
 - a. Crea un data.frame con 3 columnas:
 - i. Altura: un vector de 10 valores numéricos aleatorios entre 150 y 200.
 - ii. Peso: un vector de 10 valores numéricos aleatorios entre 50 y 100.
 - iii. Edad: un vector de 10 valores numéricos aleatorios entre 20 y 60.

2. Operaciones:

- a. Calcula el promedio de las columnas altura y peso usando lapply() o sapply().
- b. Filtra las filas donde la edad es mayor a 40 y muestra los resultados.

Parte 7: Creación de Funciones Personalizadas (10 puntos)

1. Función para Operaciones Aritméticas:

- a. Crea una función llamada operaciones_basicas() que acepte dos números y devuelva una lista con la suma, resta, multiplicación y división de esos dos números.

2. Aplicación de la Función:

- a. Utiliza la función operaciones_basicas() para calcular los resultados de operar con los números 8 y 4.

Entrega:

- La tarea debe ser entregada en un **archivo .R** donde incluyas todo el código necesario, así como comentarios que expliquen cada sección del código.
- Debes ejecutar tu código y mostrar los resultados esperados.

Fecha de Entrega:

- 24/09/2024 5:30 p.m.

Criterios de Evaluación:

- **Correctitud del código:** ¿Las operaciones y funciones están correctamente implementadas?
- **Calidad del código:** ¿El código está bien estructurado y comentado?
- **Resultados correctos:** ¿Los resultados obtenidos son correctos y completos?
- **Creatividad:** ¿El estudiante fue más allá de lo básico y agregó algún análisis o funcionalidad adicional?