**TAREA 2**

**Funciones en R**

Una función en R es un conjunto de código que se puede ejecutar durante un ciclo de un programa, es creada para aceptar uno o varios argumentos de entrada

**Estructura de una función**

nombre\_función <- function (arg1, arg2){

#Cuerpo de la función

#Retorno de la función

}

Donde:

**Nombre\_función**: es el nombre que se define el propósito de la función, debe cumplir con los requisitos de R.

**Arg1, arg2**: son los argumentos que definen la función los cuales pueden ser uno o más y deben estar separados por comas

**#cuerpo de la función**: es el código que se ejecuta cuando se nombra la función.

**#Retorno de la función**: es opcional y se utiliza la palabra ‘Return’ para devolver un resultado.

**Ejemplo**

#Definir la función para calcular el área de un cuadrado

area\_cuadrado<- c(base, altura){

area<- base\*altura

return(area)

}

#asignar valores a la función

resultado <-area\_cuadrado (5,10)

#Dar el resultado

print(resultado)

**Argumentos de una función**

Las funciones en R pueden tener un argumento o más, los cuales son valores que se pasan a la función para que se ejecute una instrucción, los argumentos pueden ser de tipo numérico, vectores, matrices, data frame, etc.

**Argumentos por valor**

Los argumentos se pasan a las funciones por valor de forma automática, es decir que cuando se nombra la función, se crea una primera copia de los argumentos y se utiliza esta copia dentro de la función, los cambios que se realizan dentro de la función no afectan a los valores fuera de la función.

En resumen, es definir la función con un argumento, asignar un valor a una variable el cual es independiente a lo que suceda dentro de la función, luego esta variable se asigna como argumento dentro de la función la cual devuelve un resultado según la función.

**Familia Apply**

Son un conjunto de funciones que permiten aplicar una función a una lista, vector, matriz, data frame y devolver un resultado en un formato especifico

**lapply()**

Se utiliza para aplicar una función a cada elemento de una lista y devolver una nueva lista del mismo tamaño que la lista original. Su sintaxis es

lapply (lista, función)

Ejemplo

Supongamos que queremos asignar la función de elevar al cuadrado los elementos de una lista

numeros <- lists(3,5,6,7) #definimos la lista de elementos

Elevar\_cuadrado <- lapply(numeros, function(x) x^2) #definimos le función para elevar al cuadrado los elementos de la lista números, los resultados se almacenan en la lista “Elevar\_cuadrados”

Primero, se define una lista de números con la función lists() que contiene los valores 3, 5, 6 y 7 y se almacena en la variable numeros.

Luego, se utiliza la función lapply() para aplicar la función anónima function(x) x^2 a cada elemento de la lista numeros. La función anónima toma un argumento x y eleva x al cuadrado. La función lapply() recorre cada elemento de la lista numeros y aplica la función anónima a cada elemento.

El resultado se almacena en la variable Elevar\_cuadrado, que es otra lista que contiene los valores resultantes de elevar al cuadrado cada elemento de la lista original. En este caso, Elevar\_cuadrado contendrá los valores 9, 25, 36 y 49, respectivamente, que son los resultados de elevar al cuadrado cada elemento de numeros.

Sapply()

La siguiente función se utiliza para aplicar una función a cada uno de los elementos de una lista, vertor o matriz y devolver un vector o una matriz. Es una función parecida a lapply pero sapply busca simplificar el resultado en un vector o matriz si es posible, si una lista tiene una misma longitud y tipo, sapply devolverá una matriz, si no tienen la misma longitud o tipo sapply va a devolver un vector. Su sintaxis es:

sapply(lista, función)

Matriz <- matrix(1:9,nrow=3)

Filas\_suma <- sapply (1:nrow(matriz), function (i) sum(matriz[i, ]))

Primero, se utiliza la función matrix() para crear una matriz llamada Matriz con los valores del 1 al 9, organizados en fila, y con 3 filas y 3 columnas.

Luego, se utiliza la función sapply() para aplicar una función anónima a cada fila de la matriz Matriz. La función anónima toma un argumento i, que representa el número de fila actual, y utiliza la función sum() para sumar los valores de la fila correspondiente.

La función sapply() recorre cada número de fila de la matriz Matriz y aplica la función anónima a cada fila. El resultado se almacena en la variable Filas\_suma, que es un vector que contiene las sumas de cada fila de la matriz original.

Vapply

La función se utiliza para aplicar una función a cada elemento de una lista, un vector o una matriz y devolver un tipo de dato. Es similar a sapply pero requiere el tipo de datos que se va a devolver, esto es funcional si se quiere asegurar que el resultado de sapply sea el tipo que se esperaba. Su sintaxis es:

vapply(lista, función, tipo\_de\_dato)

Matriz <- matrix (1:9, nrow=3)

columnas\_media <- vapply (matriz, function (x) mean (x), numeric(1))

Primero, se utiliza la función matrix() para crear una matriz llamada Matriz con los valores del 1 al 9 organizados en fila, y con 3 filas y 3 columnas.

Luego, se utiliza la función vapply() para aplicar una función anónima a cada elemento de la matriz Matriz. La función anónima toma un argumento x, que representa cada valor de la matriz, y utiliza la función mean() para calcular la media de cada columna de la matriz.

La función vapply() recorre cada elemento de la matriz Matriz y aplica la función anónima a cada elemento. El resultado se almacena en la variable columnas\_media, que es un vector que contiene las medias de cada columna de la matriz original.