




Finanzas en R


Notebook Ejercicios 01


Sebastián Egaña Santibáñez 

Nicolás Leiva Díaz 

Enlaces del profesor

 <https://segana.netlify.app>

 <https://github.com/sebaegana>

 <https://www.linkedin.com/in/sebastian-egana-santibanez/>

Ejercicio 1

1. Escriba una función que sume todos los números divisibles por 3 y 5 menores a 1000.

Solución:

```
# Inicializamos la variable para almacenar la suma
suma <- 0

# Iteramos desde 1 hasta 999 (menos de 1000)
for (numero in 1:999) {
  # Si el número es divisible por 3 o 5, lo sumamos a la variable 'suma'
```

```

    if (numero %% 3 == 0 & numero %% 5 == 0) {
      suma <- suma + numero
    }
  }

# Imprimimos el resultado
print(suma)

```

2. Identifique los operadores utilizados en su código
3. Comparelo con la misma solución en Python, y realice tres comentarios sobre las diferencias observadas

Ejercicio 2

1. Escriba una función que imprima los números desde 1 a 100 y para los números que son múltiplos de 3 imprima “TIC” en lugar del número, y para los números múltiplos de 5 imprima “TOC”, y si el número es múltiplo de 3 y 5 a la vez, imprima “TICTOC”

Solución:

```

# Function to print "Tic", "Toc", or "TicToc" based on the number
print_tic_toc <- function(number) {
  if (number %% 3 == 0 && number %% 5 == 0) {
    print("TicToc")
  } else if (number %% 3 == 0) {
    print("Tic")
  } else if (number %% 5 == 0) {
    print("Toc")
  } else {
    print(number)
  }
}

# Loop from 1 to 100 and print the result for each number
for (number in 1:100) {
  print_tic_toc(number)
}

```

Ejercicio 3

1- Haga un programa que filtre el arreglo y devuelva un arreglo con solo el nombre de sus amigos. Si un nombre tiene exactamente 4 letras, ¡puedes estar seguro que es amigo tuyo! De lo contrario, puede estar seguro de que no.

Considere el siguiente input:

```
Amigo = c("Ryan", "Kieran", "Mark", "Miguel")
```

Que debería ser:

```
Deberia = C("Ryan", "Mark")
```

Solución:

```
# Arreglo de amigos
amigos <- c("Ryan", "Kieran", "Mark", "Miguel")

# Filtrar los nombres de amigos con exactamente 4 letras
deberia <- amigos[nchar(amigos) == 4]

print(deberia)
```

Como función:

```
library(dplyr)

filtrar_amigos <- function(amigos) {
  # Filtrar los nombres de amigos con exactamente 4 letras
  amigos_filtrados <- amigos[nchar(amigos) == 4]
  return(amigos_filtrados)
}

# Arreglo de amigos
amigos <- c("Ryan", "Kieran", "Mark", "Miguel")

# Obtener el resultado utilizando la función
resultado <- filtrar_amigos(amigos)

# Imprimir el resultado
print(resultado)
```

2. Comparelo con la misma solución en Python, y realice tres comentarios sobre las diferencias observadas

Ejercicio 4

Escribe una función llamada `sumaDigitos` que retorne la suma de todos los dígitos de un número dado, por ejemplo:

5646 => 5+6+4+6 => 21

Solución:

```
sumaDigitos <- function(numero) {  
  suma <- 0  
  while (numero > 0) {  
    suma <- suma + (numero %% 10)  
    numero <- floor(numero / 10)  
  }  
  return(suma)  
}
```

Implementación:

```
# Ejemplo de uso  
numero1 <- 12345  
numero2 <- 9876  
  
resultado1 <- sumaDigitos(numero1)  
resultado2 <- sumaDigitos(numero2)  
  
print(resultado1) # Debería imprimir 15 (1 + 2 + 3 + 4 + 5)  
print(resultado2) # Debería imprimir 30 (9 + 8 + 7 + 6)
```

Ejercicio 5

Genere una función en R que realice lo siguiente:

1. Ingrese un número entre 1 y 9 desde el teclado usando la función de entrada y guárdalo en una variable llamada 'my_guess_number'

2. Genera un número aleatorio entre 1 y 9 y guárdalo en una variable llamada 'numero_secreto'
3. Compara los números e imprime un mensaje: si lo adivinó, imprime "¡Bingo, tú ganas!" y si no, imprime "Inténtalo de nuevo"

Hint: verifique que el número ingresado esté entre el rango válido

```
juego_adivinanza <- function() {  
  # 1. Ingresar un número entre 1 y 9 desde el teclado  
  my_guess_number <- as.integer(readline(prompt = "Ingresa un número entre 1 y 9: "))  
  
  # Verificar que el número ingresado está dentro del rango válido  
  if (my_guess_number < 1 || my_guess_number > 9) {  
    stop("El número debe estar entre 1 y 9.")  
  }  
  
  # 2. Genera un número aleatorio entre 1 y 9  
  numero_secreto <- sample(1:9, 1)  
  
  # 3. Compara los números e imprime un mensaje  
  if (my_guess_number == numero_secreto) {  
    print("¡Bingo, tú ganas!")  
  } else {  
    print("Inténtalo de nuevo")  
  }  
}
```

Probamos:

```
juego_adivinanza()
```