

Ejercicios N°1: Conceptos básicos

Introducción al Análisis Geoespacial CBIT202-24

La guía consta de **9** ejercicios de conceptos básicos de R, y esta diseñada para ser respondida en modo “visual”. Recuerde reemplazar todas las partes con ‘#???’ y ‘???’ con su código, y/o trabaje en la zona de ‘desarrollo’. **Esta evaluación será revisada por pares.**

Realice el desarrollo de estos ejercicios al interior del chunk correspondiente. Si necesita escribir hágalo ocupando #, esto es recomendable para aclarar qué hace su código. Puede ayudarse de inteligencia artificial para investigar sobre el funcionamiento de funciones o investigar, pero resuelva las preguntas en base a su propio conocimiento e investigación. Recuerde que puede ocupar ? o F1 para saber más información sobre las funciones, paquetes instalados y datos contenidos en R base.

```
# Por ejemplo:
```

```
# Funcion
?mean()
# Paquete
?stats()
# Datos
?iris()
```

Para acceder a los atajos del teclado puede presionar Alt + Shift + K (en Windows/Linux), Option + Shift + K (en macOS) o Help > Keyboard Shortcuts Help. No olvide que puede retroceder en cualquier cambio que realice con Ctrl + Z.

Ejercicios (21pts):

1) Dados los siguientes objetos. Cuál es la clase (class()) y el tipo (typeof()) de cada objeto ¿Ve alguna diferencia entre los resultados obtenidos con class() y con typeof()? ¿Qué significa que un objeto sea integer o double? **(2pts)**

```
objeto_1 <- "HolaMundo"
objeto_2 <- 1L
objeto_3 <- 9.2
objeto_4 <- TRUE
```

```
# Desarrollo
```

Si siente curiosidad por qué objetos tiene guardados en la memoria puede utilizar:

```
print("Objetos en memoria:")
print(ls())
print("Detalles de objetos en memoria:")
print(ls.str())
```

2) Realice las siguientes operaciones aritméticas (con: +, -, *, /, ^, %%, %/%, sqrt(), log(), pi()). **(2pts)**

```
numero_inicial <- 10.0
```

```
# divida numero_inicial por 2
```

```

obj_division <- #???

# obtenga el resto de la división de numero_inicial por 2
obj_resto <- #???

# realice una división entera de numero_inicial por 2
obj_div_entera <- #???

# multiplique obj_division por 2
obj_multiplicacion <- #???

# sume el obj_division con el obj_multiplicacion
obj_adicion <- #???

# reste el obj_multiplicacion a numero_inicial
objeto_subtraccion <- #???

# eleve el numero_inicial al cubo
obj_elevacion <- #???

# calcule la raiz cuadrada de numero_inicial
obj_raiz <- #???

# calcule el logaritmo en base 10 de numero_inicial multiplicado por pi
obj_log_por_pi <- #???

```

3) Borre de la memoria de R los objetos creados en las dos preguntas anteriores. Después borre específicamente el objeto creado en el siguiente chunk (vea la función `rm()` y sus ejemplos) (2pts).

```

# Objetos en memoria:
ls()

# Ocupe rm() para borrar todos los objetos en memoria:
rm(???)

# Borre un objeto en específico"
borrame <- '¿Podrías borrarame, porfi? No me gusta existir. Creo que tengo problemas existenciales'
borrame

rm(???)

```

4) Genere vectores que contengan (2pts):

- a- 25 veces la frase "Hola Mundo".
- b- Los números pares entre 0 y 100, incluyendo 100.
- c- 1000 números pertenecientes a una distribución normal de media 10 y desviación estándar 2.
- d- La media y desviación estándar reales del vector anterior (c).
- e- 10 números aleatorios entre 0 y 10
- f- La multiplicación de los valores del vector anterior (e) por 5
- g- Al vector anterior (f) agréguele el valor 77
- h- Genere la suma acumulativa de los valores de un vector con los números enteros del 2 al 20 (vea `cumsum()`)

```

a <- #???
print(a)

b <- #???
print(b)

c <- #???
print(c)

d <- #???
print(d)

e <- #???
print(e)

f <- #???
print(f)

g <- #???
print(g)

h <- #???
print(h)

```

5) Genere dos vectores, un vector alpha con los números del 1 al 10 y un vectos beta de largo 5 con nombres de frutas. En base a estos responda (5pts):

a- ¿Cómo puedo saber el largo de cada vector?

b- ¿Cómo puedo imprimir solo el primer elemento de cada vector? ¿Y el último?

c- ¿Cómo puedo remplazar el ultimo elemento del vector beta con otra fruta?

d- ¿Cómo puedo obtener la media, desviación estandar, varianza, valor máximo y valor mínimo del vector alpha?

e- ¿Se pueden combiar estos dos vectores en un único objeto? Demuéstrelo. ¿En que se diferencia una lista de un vector?

f- ¿Con qué función puedo ver un resumen estadístico de estos vectores?

g- ¿Cómo puedo ordenar los valores de ambos vectores de menor a mayor o de la A a la Z? ¿Y de la manera contraria? (Pista: revise la función sort() y sus argumentos)

h- ¿Cómo puedo mostrar los valores superiores o iguales a 8 en el vector alpha ocupando la función which y []?

i- ¿Cómo puedo generar con sample() una muestra de tamaño 5 con remplazo de alpha y beta? 1- ¿Cómo, con sample(), puedo revolver aleatoriamente el orden de los datos del vector alpha y beta?

```

alpha <- #???
beta <- #???

# Desarrollo:
# a:

# b:

# c:

```

```
# d:
```

```
# e:
```

```
# f:
```

```
# g:
```

```
# h:
```

```
# i:
```

```
# l:
```

6) El siguiente código contiene 8 errores, corríjalos (2pts):

```
# Desarrollo
```

```
# Error 1
```

```
mi_variable1 <- 10
```

```
print(mi_variable1)
```

```
# Error 2
```

```
# mi_vector_numerico debe ser de clase numerico
```

```
mi_vector_numerico <- (1, 2 3, 4, "5", 6, 7, 8, 9, 10, NA)
```

```
print(class(mi_vector_numerico))
```

```
# Error 3
```

```
# mi_media debe ser un numero
```

```
mi_media <- mean(mi_vector_numerico, na.rm = FALSE)
```

```
print(mi_media)
```

```
# Error 4
```

```
# Las variables a y b son iguales? El resultado debería ser TRUE
```

```
a <- 10
```

```
b <- 10
```

```
print(paste("¿A es igual a b?:", a == b))
```

```
# Error 5
```

```
vector_caracteres <- c('uno', 'dos', 'tres', 'cuatro', 'cinco')
```

```
print(vector_caracteres[-1])
```

```
# Error 6
```

```
valor_maximo <- Max(runif(100, 0, 50))
```

```
# Error 7
```

```
# quiero el ultimo valor de v
```

```
v <- 1:5
```

```
v[6]

# Error 8
print("Hola mundo")
```

7) Genere los objetos numéricos aa, bb, cc y dd de tal manera que se cumpla que:

- aa es mayor que bb
- bb es igual a cc
- aa es distinto a bb
- aa es menor o igual a dd

Demuéstrelo ocupando comparaciones lógicas de tal manera que el resultado sea TRUE (2pts).

```
aa <- #???
bb <- #???
cc <- #???
dd <- #???
```

Desarrollo:

8) Use paste() para escribir el siguiente texto “hola estrellitas el mundo les dice hola”, con un espacio de separación entre palabras. Repita el proceso con las siguientes separaciones: “/” y “_”. Después ocupe paste0() ¿en qué se diferencia paste() y paste0()? (2pts)

```
# Ejemplo
texto <- paste('Hola', 'Mundo', sep = " ")
print(texto)
```

Desarrollo:

9) Diga cuáles son las variables del data frame “trees” (contenido en R base), en qué unidades de medida estan cada una y cuál es la fuente de los datos. Ocupe la función colnames() para saber el nombre de las columnas de data frame y nrow() para saber la cantidad de observaciones.

Ocupe las funciones glimpse(), view() y summary() sobre el data frame y vea que ocurre. Tenga en cuenta que estas funciones son del paquete tidyverse (cárguelo).

Además, usé [] y comparaciones lógicas para saber a través de un vector booleano (TRUE/FALSE) cuáles árboles son mayores a 80. (2pts)

```
# Data frame de arboles:
print(trees)
# Desarrollo:
```

Si siente curiosidad por los sets de datos contenidos en R base puede correr la siguiente línea de código para ver todos los sets de datos.

```
data(package = 'datasets')
```

Bonus: Genere una función que, al entregarle los valores a y b, siendo estos los catetos de un triángulo rectángulo, calcule el valor de la hipotenusa. **(0pts)**

```
# Desarrollo:
FunPitagoras <- function(){
  ###
}

FunPitagoras(3,4)
```