

# Laboratorio 1

Javier de San Jose Recinos

2023-05-05

## Parte 1

1) ¿Cómo generaría un arreglo aleatorio de 250 elementos con los colores primarios?

```
colores_primarios <- c("rojo", "azul", "amarillo")
vector_aleatorio <- sample(1:3, 250, replace = TRUE)
arreglo_colores_primarios <- colores_primarios[vector_aleatorio]
arreglo_colores_primarios
```

```
## [1] "amarillo" "rojo" "rojo" "rojo" "rojo" "azul"
## [7] "azul" "rojo" "amarillo" "amarillo" "amarillo" "azul"
## [13] "amarillo" "amarillo" "amarillo" "amarillo" "azul" "azul"
## [19] "amarillo" "rojo" "azul" "azul" "rojo" "rojo"
## [25] "azul" "amarillo" "amarillo" "rojo" "azul" "amarillo"
## [31] "rojo" "rojo" "rojo" "rojo" "amarillo" "rojo"
## [37] "azul" "azul" "rojo" "azul" "amarillo" "rojo"
## [43] "azul" "rojo" "azul" "rojo" "rojo" "amarillo"
## [49] "azul" "rojo" "amarillo" "amarillo" "amarillo" "azul"
## [55] "amarillo" "rojo" "rojo" "rojo" "rojo" "azul"
## [61] "azul" "amarillo" "azul" "azul" "amarillo" "azul"
## [67] "rojo" "azul" "rojo" "amarillo" "amarillo" "amarillo"
## [73] "rojo" "rojo" "azul" "azul" "amarillo" "rojo"
## [79] "amarillo" "rojo" "amarillo" "rojo" "amarillo" "rojo"
## [85] "amarillo" "azul" "rojo" "amarillo" "amarillo" "azul"
## [91] "rojo" "azul" "azul" "azul" "rojo" "azul"
## [97] "rojo" "amarillo" "rojo" "amarillo" "amarillo" "rojo"
## [103] "rojo" "amarillo" "azul" "azul" "azul" "rojo"
## [109] "amarillo" "azul" "rojo" "rojo" "rojo" "amarillo"
## [115] "rojo" "amarillo" "rojo" "azul" "amarillo" "azul"
## [121] "rojo" "azul" "rojo" "amarillo" "azul" "amarillo"
## [127] "rojo" "amarillo" "azul" "azul" "azul" "amarillo"
## [133] "amarillo" "amarillo" "azul" "azul" "rojo" "rojo"
## [139] "rojo" "azul" "rojo" "rojo" "amarillo" "rojo"
## [145] "amarillo" "amarillo" "amarillo" "azul" "rojo" "amarillo"
## [151] "amarillo" "amarillo" "rojo" "amarillo" "azul" "azul"
## [157] "azul" "amarillo" "azul" "rojo" "amarillo" "rojo"
## [163] "amarillo" "rojo" "amarillo" "rojo" "amarillo" "rojo"
## [169] "rojo" "rojo" "azul" "azul" "amarillo" "azul"
## [175] "azul" "azul" "amarillo" "amarillo" "rojo" "amarillo"
## [181] "amarillo" "azul" "amarillo" "amarillo" "azul" "amarillo"
## [187] "azul" "azul" "amarillo" "amarillo" "rojo" "amarillo"
## [193] "amarillo" "amarillo" "rojo" "azul" "azul" "azul"
## [199] "rojo" "azul" "rojo" "rojo" "azul" "azul"
```

```
## [205] "rojo"      "azul"      "rojo"      "azul"      "azul"      "amarillo"
## [211] "azul"      "amarillo"  "rojo"      "rojo"      "azul"      "rojo"
## [217] "amarillo"  "amarillo"  "rojo"      "azul"      "rojo"      "rojo"
## [223] "rojo"      "rojo"      "amarillo"  "rojo"      "amarillo"  "azul"
## [229] "amarillo"  "amarillo"  "rojo"      "rojo"      "azul"      "rojo"
## [235] "rojo"      "amarillo"  "amarillo"  "rojo"      "azul"      "rojo"
## [241] "rojo"      "amarillo"  "amarillo"  "rojo"      "rojo"      "amarillo"
## [247] "amarillo"  "azul"      "amarillo"  "amarillo"
```

- 2) Dado un arreglo de 120000 elementos de números enteros entre 40 y 70, ¿Cómo haría para obtener la desviación estándar de aquellos números que son mayores a 55 y menores a 64?

You can also embed plots, for example:

```
arreglo <- sample(40:70, 120000, replace = TRUE)
sub_arreglo <- arreglo[arreglo > 55 & arreglo < 64]
desv_est <- sd(sub_arreglo)
desv_est
```

```
## [1] 2.290755
```

- 3) Suponga que se almacena en una variable un arreglo con todos los números de carnet de los estudiantes del Instituto de Investigación de Operaciones de la universidad en forma de string, los carnets tiene al inicio dos caracteres que identifican el año en el que la persona entró a la universidad por ejemplo: 19004528 hacer referencia a una persona que ingreso en el año 2019, 06009878 hace referencia a una persona que ingreso que el año 2006, ¿Cómo haría para contar la cantidad de alumnos por año?

```
numeros_carnet <- c("19004528", "19004529", "06009878", "06009879", "21225130", "22000123", "22000124",
años_ingreso <- substr(numeros_carnet, 1, 2)
cantidad_por_anio <- table(años_ingreso)
cantidad_por_anio
```

```
## años_ingreso
## 06 19 21 22
##  2  2  1  4
```

- 4) Define el vector  $y = (1, 3, 5, 7)$  utilizando la función `c()`. ¿Cómo lo harías con la función `seq()`?

```
s <- seq(from = 1, to = 7, by = 2)
s
```

```
## [1] 1 3 5 7
```

- 5) Si `b <- list(a=1:10, c="Hola", d="XX")`, escriba una expresión en R que devuelva todos los elementos de la lista excepto los elementos 4,6,7,8 del vector a.

```
b <- list(a=1:10, c="Hola", d="XX")
b$a <- b$a[-c(4,6,7,8)]
b
```

```
## $a
## [1]  1  2  3  5  9 10
##
## $c
## [1] "Hola"
##
## $d
## [1] "XX"
```

6) Dado los siguientes vectores, muestre el código en R necesario para declarar y calcular el promedio, desviación estándar, mediana, moda y sumatoria total para los siguientes vectores: a.x=(8,7,6,5)

b. y = (3,3, 3, 3, 3, 3, 3, 3, 2, 2)

c. z = (1, 1.75, 2.5, 3.25, 4)

```
x <- c(8, 7, 6, 5)
y <- c(3,3, 3, 3, 3, 3, 3, 3, 2, 2)
z <- c (1, 1.75, 2.5, 3.25, 4)

# Promedio
mean(x)

## [1] 6.5
mean(y)

## [1] 2.8
mean(z)

## [1] 2.5
# Desviación estándar
sd(x)

## [1] 1.290994
sd(y)

## [1] 0.421637
sd(z)

## [1] 1.185854
# Mediana
median(x)

## [1] 6.5
median(y)

## [1] 3
median(z)

## [1] 2.5
# Moda (utilizando la función 'mfv' definida previamente)
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
getmode(x)

## [1] 8
getmode(y)

## [1] 3
getmode(z)
```

```
## [1] 1
```

```
# Sumatoria total  
sum(x)
```

```
## [1] 26
```

```
sum(y)
```

```
## [1] 28
```

```
sum(z)
```

```
## [1] 12.5
```

7) Sabemos que para sumar vectores estos deben tener la misma longitud. Sin embargo R trabaja de manera distinta. Defina los vectores  $x = (1, 2, 3, 4, 5, 6)$ ,  $y = (7, 8)$ ,  $z = (9, 10, 11, 12)$ . Calcula:

- $x + x$
- $x + y$ .
- Responda ¿Qué ha hecho R?

```
x <- c(1, 2, 3, 4, 5, 6)  
y <- c(7, 8)  
z <- c(9, 10, 11, 12)
```

```
x + x
```

```
## [1] 2 4 6 8 10 12
```

```
x+y
```

```
## [1] 8 10 10 12 12 14
```

En la operación  $x + y$ , R recicla los elementos de  $y$  para que tenga la misma longitud que  $x$ , es decir, convierte  $y$  en  $(7, 8, 7, 8, 7, 8)$  y luego suma los elementos correspondientes de  $x$  e  $y$ . Esto produce el vector resultante  $(8, 10, 10, 12, 12, 14)$ .

8) Dada la variable `string <- "Hola Mundo"`, escriba una instrucción en R que devuelva la siguiente salida:

```
palabra <- "Hola Mundo"  
palabras <- strsplit(palabra, " ")  
palabras
```

```
## [[1]]
```

```
## [1] "Hola" "Mundo"
```

## Segunda Parte