

Ingeniería en sistemas computaciones

Datos masivos



Practica 2

Alumnos:

Marquez Millan Seashell Vanessa

Castro Cebreros Alejandro

```
#20 funciones practica 2
```

```
1-#dibuja la gráfica de densidad del núcleo.
```

```
set.seed(13531)
```

```
# Create random numeric
```

```
data
```

```
x <- rnorm(1000)
```

```
plot(density(x))
```

```
2-.cat(x) #Esta función escribe texto y variables en la salida.
```

```
x <- 2
```

```
y <- 4
```

```
cat(x,"elevado a",y,"es",x ^ y,"\n")
```

```
3-.invisible(x) #Oculta x en la función return de forma que no se ve el resultado
```

```
suma <- function(a, b) {
```

```
  s <- a + b
```

```
  return(invisible(s))
```

```
}
```

```
suma(5,4)
```

```
x <- suma(5,4)
```

```
x
```

```
4-. #convierte valores numéricos en rangos factoriales.
```

```
x <- 1:10
```

```
# Create example vector
```

```
x
```

```
x_cut <- cut(x, breaks = c(0, 4, 6, 7, 10)) # Apply cut()
```

```
x_cut
```

```
5-.trunc(x)
```

```
trunc(1.999999)
```

```
6-.abline ( h = 1 ) #agrega líneas rectas para trazar.
```

```
set.seed(9764355)
```

```
x <- rnorm(1000)
```

```
y <- rnorm(1000) + 0.4 * x
```

```
plot ( x, y )
```

```
plot(x, y)
abline(h = 1.3)
```

7-.aggregate(x) #calcula estadísticas resumidas de subgrupos de un conjunto de datos.

```
data <- data.frame(x1 = 1:5,
                  x2 = 2:6,
                  x3 = 1,
                  group = c("A", "A", "B", "C", "C"))

data
```

```
aggregate(x = data[, colnames(data) != "group"],
          by = list(data$group),
          FUN = mean)
```

8-.all(x) #comprueba si todos los valores de un vector lógico son VERDADEROS.

```
x1 <- c(1, 5, 3, - 3, 5, - 7, 8) # Example vector
x1
all(x1 < 0)
```

9-. beep() #reproduce el sonido al final de una secuencia de comandos R.

```
install.packages("beepR")
library(beepR)
```

```
beep()
```

10-.boxplot(x) #crea un diagrama de caja

```
set.seed(8642) # Create
random data
x <- rnorm(1000)
boxplot(x)
```

11-.between(x1, left1, right1) #prueba si un valor numérico cae dentro de un rango especificado.

```
install.packages("dplyr") # Install dplyr package
library("dplyr") # Load dplyr package
```

```
x1 <- 5 # Define value
left1 <- 3 # Define lower range
right1 <- 8 # Define upper range
between(x1, left1, right1)
```

12-. bind_rows(data1, data2) # enlaza columnas y vectores.

```
data1 <- data.frame(x1 = 1:5, # Create three data frames
                  x2 = letters[1:5])
data2 <- data.frame(x1 = 0,
                  x3 = 5:9)
data3 <- data.frame(x3 = 5:9,
                  x4 = letters[5:9])
```

```
install.packages("dplyr") # Install dplyr package
```

```

library("dplyr") # Load dplyr package

bind_rows(data1, data2)

13-.round(x2) #redondee el valor numérico hasta el siguiente número entero superior.
x2 <- 1.9
x3 <- - 1.1
round(x2)
round(x3)

14-.tolower(x) #Cambia a mayusculas o minusculas
x <- "Example"
tolower(x) # Convert to lower case letters

toupper(x) # Convert to upper case letters

15-. coalesce(x, 999) #redondee el valor numérico hasta el siguiente número entero superior.
install.packages("dplyr") # Install and load dplyr package
library("dplyr")
x <- c(2, 1, NA, 5, 3, NA)
coalesce(x, 999)

16-.rowMeans(data) # la media de cada columna de un marco, matriz o matriz de datos numéricos.
set.seed(1234) # Set seed
data <- data.frame(matrix(round(runif(12, 1, 20)), # Create example
data
nrow = 3, ncol = 4))

data
colSums(data)
colMeans(data)
rowMeans(data)

17-. #Version de switch case
x1 <- 1:6 # Create first vector
x1 # Print first vector

x2 <- letters[1:6] # Create second vector
x2

install.packages("dplyr") # Install & load dplyr
package
library("dplyr")

new_dplyr <- case_when(x1 < 3 ~ "Group 1", # Applying case_when
x2 %in% letters[2:5] ~ "Group 2",
TRUE ~ "Group 3")
new_dplyr # Print output

```

18-.

```
install.packages("dplyr")          # Install and load dplyr package
library("dplyr")
```

```
c(5, 7, 1, 5, 1)
combine(5, 7, 1, 5, 1)
```

19-.#devuelve la clasificación de cada valor de nuestro vector de entrada

```
x <- c(4, 1, 5, 2, 3, 3)
row_number(x)
```

20-.#calcula la suma acumulativa de los elementos de un vector numérico.

```
set.seed(456654)          # Set seed for reproducibility
x <- round(runif(10, 1, 9)) # Create example vector
x
cumsum(x)
```