**Sesión 7 *R Data Structure***

Contenido: vectors, list, factors, matrices, array, Data frame, missing values

R. **Actividad 7**

Aplicaciones en R utilizando data frame de esta sesión

Estructura de datos en R

Vectores

Los vectores son arreglos ordenados en los cuales se puede almacenar información numérica (Cuantitativa), alfanumérico (Cualitativa), booleano (True, false), o combinaciones de los anteriores. En R pueden ser creados por medio de la instrucción c()que significa concatenar o por medio de las funciones rep(), seq() y :que permiten la creación de patrones.

Presentamos a continuación ejemplos de creación de vectores.

vector <- **c**(1, 5, 84, 9, 0)

vector

## [1] 1 5 84 9 0

vector\_1 <- **c**("a", "b", "c")

vector\_1

## [1] "a" "b" "c"

vector\_2 <- **c**(T, F, F, T)

vector\_2

## [1] TRUE FALSE FALSE TRUE

vector\_3 <- **c**(2, 3, "c")

vector\_3

## [1] "2" "3" "c"

vector\_4 <- 1**:**5

vector\_4

## [1] 1 2 3 4 5

*# seq(inicio, fin, salto)*

vector\_5 <- **seq**(1,5,0.4)

vector\_5

## [1] 1.0 1.4 1.8 2.2 2.6 3.0 3.4 3.8 4.2 4.6 5.0

vector\_6 <- **rep**(2,6)

vector\_6

## [1] 2 2 2 2 2 2

La función generadora de niveles, gl(k,n) donde k es el número de niveles o clases y n el número de repeticiones para cada nivel.

v1 <- **gl**(2,4, labels = **c**("Hombre", "Mujer"))

v1

## [1] Hombre Hombre Hombre Hombre Mujer Mujer Mujer Mujer

## Levels: Hombre Mujer

Para la contrucción un vector vacio

v <- **vector**()

Preguntarle a R si el objeto es un vector

**is.vector**(v)

## [1] TRUE

Transformar un objeto en un vector

**as.vector**(v)

## logical(0)

Para obtener el número de elementos del vector

**length**(v1)

## [1] 8

Estracción de elementos de un vector

Podemos acceder a los elementos de un vector por medio de índices y el signo [].

vector\_1[2]

## [1] "b"

vector\_4[5]

## [1] 5

Eliminar elementos de un vector

Podemos eliminar elementos de un vector por medio de índices y el signo menos.

vector\_4

## [1] 1 2 3 4 5

vector\_4[**-**5]

## [1] 1 2 3 4

Matrices y arrays

Las matrices son arreglos rectangulares de filas y columnas con información numérica, alfanumérica o lógica que se encuentra indexada. Para la construcción de una matriz en R se usa la función matrix(). Por otro lado los arrays tienen la misma estructura solo que pueden contener varias dimensiones, estos se construyen por medio de la función array().

Los métodos asociados a la función matriz y a la función array se muestran a continuación

**matrix**(data , nrow , ncol , byrow )

**array**(data, dim )

* data: Datos que forman la matriz.
* nrow: Número de filas.
* ncol: Número de columnas.
* byrow: Rellenar la información por filas.
* dim: Dimensiones del array

Algunos ejemplos de matrices y array son:

matriz <- **matrix**(**c**(1,2,36,9,5,2),ncol = 3, byrow = T)

matriz

## [,1] [,2] [,3]

## [1,] 1 2 36

## [2,] 9 5 2

arreglo <- **array**(**c**(**rep**(1,5),**rep**(3,5),2,6), **c**(2,3,2))

arreglo

## , , 1

##

## [,1] [,2] [,3]

## [1,] 1 1 1

## [2,] 1 1 3

##

## , , 2

##

## [,1] [,2] [,3]

## [1,] 3 3 2

## [2,] 3 3 6

Conocer el número de elementos en la matriz

m <- **matrix**(data=5, nr=2, nc=2)

**length**(m)

## [1] 4

Conocer la dimensión de la matriz

**dim**(m)

## [1] 2 2

Contruir una matriz vacia

m1 <- **matrix**()

m1

## [,1]

## [1,] NA

Asignar nombres a las columnas y a las filas.

**rownames**(m) = **c**("x1","x2")

**colnames**(m) = **c**("x1","x2")

m

## x1 x2

## x1 5 5

## x2 5 5

Reasignando los valores para los elementos de una matriz

Extraer elementos de una matriz y un array

Podemos acceder a los elementos de una matriz o un array por medio de índices y el signo [].

matriz[2,1]

## [1] 9

arreglo[2,1,1]

## [1] 1

Reasignando los valores para los elementos de una matriz

m[1,1] = 1

m[2,2] = 4

m

## x1 x2

## x1 1 5

## x2 5 4

Recorrer los elementos de la matriz.

m[3]

## [1] 5

Eliminar elemntos de una matriz

m[2] <- NA

m

## x1 x2

## x1 1 5

## x2 NA 4

Determinante e inversa de una matriz

m[2] <- 7

Calculo del deterinante de una matriz

**det**(m)

## [1] -31

Calculo de la inverda de una matriz.

**solve**(m)

## x1 x2

## x1 -0.1290323 0.16129032

## x2 0.2258065 -0.03225806

Data frame

Los data frame son estructuras de datos que permiten agrupar vectores diferentes tipos de datos (numérica, categórica, booleana) en un mismo objeto, pero que debe cumplir la condición de que todos los vectores deben tener la misma longitud. En R los data frame son generados por medio de la función data.frame().

Algunos ejemplos de data frame en R.

vec\_1 <- 1**:**5

vec\_2 <- **c**("a", "e", "i", "o", "u")

marco.datos <- **data.frame**(Numeros = vec\_1, Letras = vec\_2)

marco.datos

## Numeros Letras

## 1 1 a

## 2 2 e

## 3 3 i

## 4 4 o

## 5 5 u

Extraer elementos de una data frame

Podemos acceder a los elementos de un data frame por medio de índices y el signo []. Por lo general se especifican tanto la fila como la columna a la cual pertenece el data a extraer eso se hace por medio de la sintaxis data.frame[#fila,#Columna]

Algunos ejemplos

marco.datos[,1] *# Se extrae solo la columna 1*

## [1] 1 2 3 4 5

marco.datos[2,2]*# Se extrae elemento de la fila 2 columna 2*

## [1] e

## Levels: a e i o u

x <- 1**:**20

y <- **rep**(**c**("a", "b", "c", "d"),5)

df <- **data.frame**(x,y)

df**$**y

## [1] a b c d a b c d a b c d a b c d a b c d

## Levels: a b c d

df[,"y"]

## [1] a b c d a b c d a b c d a b c d a b c d

## Levels: a b c d

Listas

Las listas son estructuras muy usadas para almacenar objetos donde cada uno cada uno puede tener un estructura distinta. En R las listas se pueden crear por medio de la función list()

A continuación se presentan ejemplos sobre listas en R.

vec\_1 <- 1**:**5

vec\_2 <- **c**("a", "e", "i", "o", "u")

marco.datos <- **data.frame**(Numeros = vec\_1, Letras = vec\_2)

matriz <- **matrix**(**c**(1,2,36,9,5,2),ncol = 3, byrow = T)

arreglo <- **array**(**c**(**rep**(1,5),**rep**(3,5),2,6), **c**(2,3,2))

lista <- **list**(marco.datos, matriz, arreglo)

Extraer elemento de una lista

lista[2]

## [[1]]

## [,1] [,2] [,3]

## [1,] 1 2 36

## [2,] 9 5 2

marco.datos[[1]]

## [1] 1 2 3 4 5