



Dia 1. Introducción al R básico

¿Qué es R?

Principios de la investigación reproducible.

Tipos y estructuras de datos en R.

Universidad de Guadalajara, CUCSH–CUCBA
Viacheslav Shalisko

16–24.07.2019

Lenguaje R

Clases de *objetos* mas usados en R

1. Estructuras de datos

- a) Básicas (~5)
- b) Adicionales (definidos por módulos)

2. Funciones

- a) Básicas (parte del núcleo R)
- b) Externas (parte de los módulos)

3. Estructuras de control

R incluye varios modelos de objetos, pero esta tema es relevante para desarrolladores de módulos

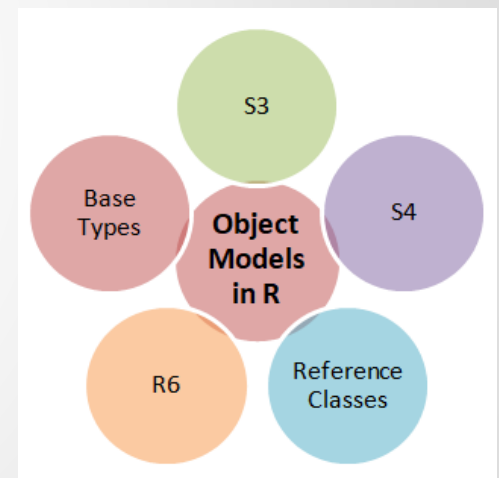


Figura de <http://www.ciaburro.it/why-is-oop-in-r-so-confusing/>

Nota 1: usualmente los objetos en R suelen tener un nombre (identificador)

Nota 2: existe una excepción de esta regla – las funciones anónimas

Estructuras de datos en R



Vector es la estructura de datos más simple

Trata de un conjunto ordenado de elementos de mismo tipo

Los tipos de elementos pueden ser:

- a) Números enteros (numeric integer, 32 bits con signo)
- b) Números reales de doble precisión (numeric double, de 64 bits)
- c) Valores lógicos (logical)
- d) Cadenas de texto (character)
- e) Factores (factor, tipo de datos compuesto)

Estructuras de datos en R

Ejemplo: vectores numéricos

```
a <- 1:10
```

```
a
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
b <- c(1.0, -3.4, 2, 140.1)
```

```
b
```

```
## [1] 1.0 -3.4 2.0 140.1
```

```
typeof(a)
```

```
## [1] "integer"
```

```
typeof(b)
```

```
## [1] "double"
```

Ejemplo: vectores de caracteres y lógicos

```
c <- c("lunes", "martes", "miercoles", "jueves", "viernes")
```

```
c
```

```
## [1] "lunes" "martes" "miercoles" "jueves" "viernes"
```

```
d <- c(TRUE, FALSE, FALSE, TRUE)
```

```
d
```

```
## [1] TRUE FALSE FALSE TRUE
```

```
typeof(c)
```

```
## [1] "character"
```

```
typeof(d)
```

```
## [1] "logical"
```

Estructuras de datos en R

Coerción de tipos es el proceso de transformación forzada de un tipo a otro

La coerción de tipos se realiza de los tipos de datos más restrictivos a los más flexibles

Existe la coerción **implícita** y **explícita**

Para la coerción explícita se usan funciones de la familia `as.` como:
`as.integer()`,
`as.numeric()`,
`as.character()`,
`as.logical()`

Secuencia de coerción

lógico -> entero -> numérico -> cadena de texto

```
c(a, b)
```

```
## [1] 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 1.0
## [12] -3.4 2.0 140.1
```

```
c(a,b,c)
```

```
## [1] "1" "2" "3" "4" "5"
## [6] "6" "7" "8" "9" "10"
## [11] "1" "-3.4" "2" "140.1" "lunes"
## [16] "martes" "miercoles" "jueves" "viernes"
```

```
c(a,b,c,d)
```

```
## [1] "1" "2" "3" "4" "5"
## [6] "6" "7" "8" "9" "10"
## [11] "1" "-3.4" "2" "140.1" "lunes"
## [16] "martes" "miercoles" "jueves" "viernes" "TRUE"
## [21] "FALSE" "FALSE" "TRUE"
```

```
c(a,d)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 1 0 0 1
```

Estructuras de datos en R

Vector de factores es la estructura de datos compuesta muy utilizada.

En breve, es un vector numérico con las etiquetas asociadas.

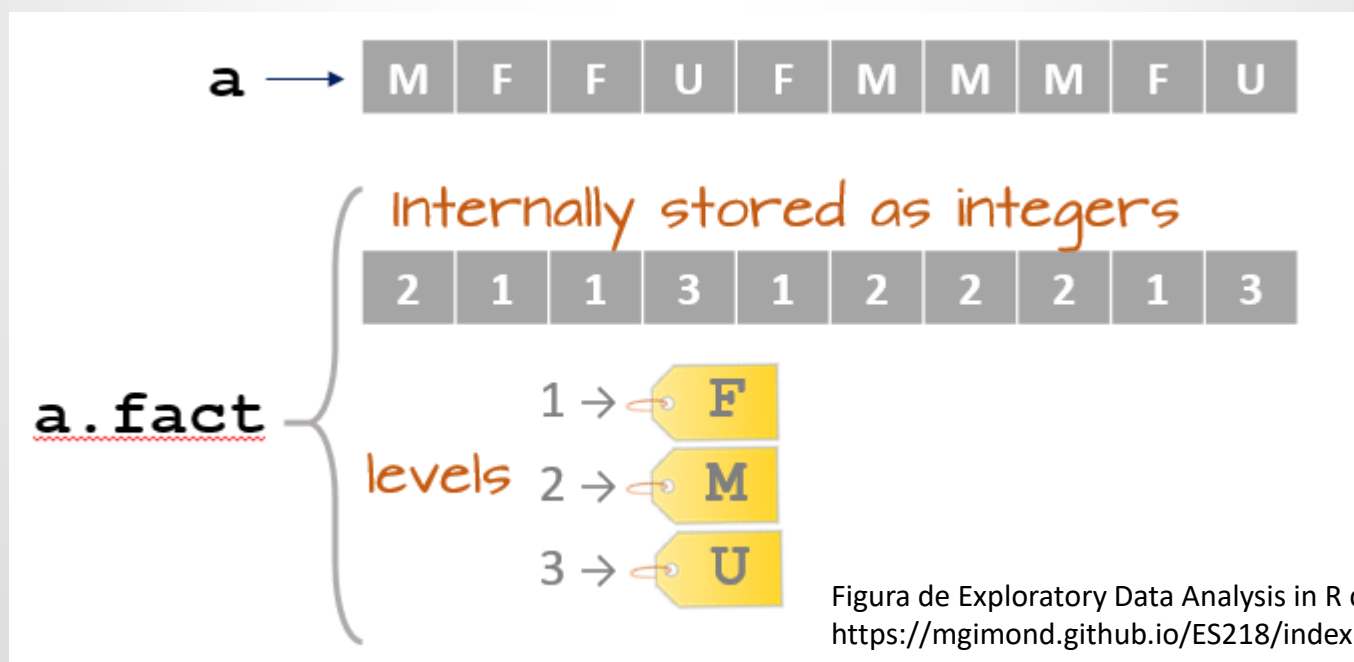


Figura de Exploratory Data Analysis in R course
<https://mgimond.github.io/ES218/index.html>

Estructuras de datos en R

Ejemplo: definición de un vector de factores

Coerción a factores:
`as.factor()`

Lista de etiquetas:
`levels()`

```
e <- c("masculino", "femenino", "masculino", "neutro", "femenino")
typeof(e)
```

```
## [1] "character"
```

```
e.fact <- as.factor(e)
levels(e.fact)
```

```
## [1] "femenino" "masculino" "neutro"
```

```
e.fact
```

```
## [1] masculino femenino masculino neutro      femenino
## Levels: femenino masculino neutro
```

```
typeof(e.fact)
```

```
## [1] "integer"
```

```
class(e.fact)
```

```
## [1] "factor"
```

Estructuras de datos en R

matrix

12.3	0.1
3.0	5.2
5.01	3.0
2.3	0.1

Matrices son las estructuras rectangulares bidimensionales

Pueden estar conformados por elementos de un solo tipo (mismos tipos que en vectores, excepto factores)

Arrays son similares a matrices pero pueden tener mas de dos dimensiones

Estructuras de datos en R

Ejemplo: definición de matriz

Definición de matrices:

`matrix(data, nrow = x, ncol = y)`

Número de dimensiones:

`dim()`

Combinación de matrices:

`cbind()`

`rbind()`

```
cbind(m1,m2)
```

```
##      [,1] [,2] [,3] [,4]      [,5]      [,6]      [,7]
## [1,]    1    4    7   10 4.643617 9.8635540 9.129943
## [2,]    2    5    8   11 9.326838 1.8079605 7.103164
## [3,]    3    6    9   12 6.601292 0.6344612 7.829705
```

```
m1 <- matrix(1:12, nrow = 3, ncol = 4)
m1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```

```
m2 <- matrix(runif(9,0,10), nrow = 3, ncol = 3)
m2
```

```
##      [,1]      [,2]      [,3]
## [1,] 9.9285730 3.642612 4.666548
## [2,] 0.9590461 9.656692 4.639480
## [3,] 4.3327996 9.009836 8.972147
```

```
dim(m1)
```

```
## [1] 3 4
```

```
dim(m2)
```

```
## [1] 3 3
```

Estructuras de datos en R

dataframe

x	y
12.3	ace
3	tea
5.01	oil
2.3	tree

Los **dataframes** son las estructuras rectangulares bidimensionales similares a las tablas.

A diferencia de matrices, las columnas pueden tener datos de distintos tipos básicos.

Un **dataframe** esta compuesto por vectores (columnas).

El número de elementos en cada columna es fijo e igual a número de líneas en **dataframe**.

Estructuras de datos en R

La definición de dataframe se realiza por la función `data.frame()`

Es posible realizar una coerción de matriz a dataframe por medio de `as.data.frame()`

Las columnas y líneas de un dataframe pueden tener nombres

Estos se asignan y se acceden por medio de `names()` y `row.names()`

Igual que en matrices

Número de dimensiones:
`dim()`

Combinación de dataframes:
`cbind()`
`rbind()`

Estructuras de datos en R

Ejemplo: definición de un dataframe

```
ciudades.df <- data.frame(  
  "id" = 1:4,  
  "nombre" = c("Ciudad de México (ZMCM)", "Guadalajara (AMG)", "Monterrey (ZMM)", "León"),  
  "tipo" = as.factor(c("megaciudad", "ciudad", "ciudad", "ciudad")),  
  "poblacion" = c(20.40, 4.75, 4.69, 1.24),  
  "latitud" = c(19.428, 20.667, 25.6714, 21.129),  
  "longitud" = c(-99.128, -103.392, -100.309, -101.674),  
  stringsAsFactors = FALSE  
)
```

ciudades.df

##	id	nombre	tipo	poblacion	latitud	longitud
## 1	1	Ciudad de México (ZMCM)	megaciudad	20.40	19.4280	-99.128
## 2	2	Guadalajara (AMG)	ciudad	4.75	20.6670	-103.392
## 3	3	Monterrey (ZMM)	ciudad	4.69	25.6714	-100.309
## 4	4	León	ciudad	1.24	21.1290	-101.674

Estructuras de datos en R

Ejemplo: propiedades de un dataframe

```
dim(ciudades.df)
```

```
## [1] 4 6
```

```
names(ciudades.df)
```

```
## [1] "id"      "nombre"  "tipo"    "poblacion" "latitud"  "longitud"
```

```
row.names(ciudades.df)
```

```
## [1] "1" "2" "3" "4"
```

```
str(ciudades.df)
```

```
## 'data.frame':    4 obs. of  6 variables:
## $ id      : int  1 2 3 4
## $ nombre  : chr  "Ciudad de México (ZMCM)" "Guadalajara (AMG)" "Monterrey (ZMM)" "León"
## $ tipo    : Factor w/ 2 levels "ciudad","megaciudad": 2 1 1 1
## $ poblacion: num  20.4 4.75 4.69 1.24
## $ latitud  : num  19.4 20.7 25.7 21.1
## $ longitud : num  -99.1 -103.4 -100.3 -101.7
```

Estructuras de datos en R

list

x	y
12.3	ace
3	tea
5.01	oil
2.3	tree

3

$Y \sim x - 1$

some
text

Listas son las estructuras unidimensionales conformados por elementos heterogéneos (que pueden ser de diferentes tipos y clases).

Los elementos de las listas pueden tener distintas dimensiones e incluso ser otras listas.

Lista solo tiene una sola dimensión que se puede acceder por medio de `length()`

Estructuras de datos en R

Ejemplo: definición de una lista

```
mi_vector <- 1:10
mi_factor <- as.factor(c("F1","F2"))
mi_matriz <- matrix(1:6, nrow = 2)
mi_df      <- data.frame("var_num" = 0:2, "var_text" = c("uno", "dos", "tres"))

mi_lista <- list("un_vector" = mi_vector, "un_factor" = mi_factor, "una_matriz" = mi_matriz, "un_df" = mi_df)
mi_lista
```

```
## $un_vector
## [1] 1 2 3 4 5 6 7 8 9 10
##
## $un_factor
## [1] F1 F2
## Levels: F1 F2
##
## $una_matriz
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
##
## $un_df
##   var_num var_text
## 1      0      uno
## 2      1      dos
## 3      2      tres
```

```
str(mi_lista)
```

```
## List of 4
## $ un_vector : int [1:10] 1 2 3 4 5 6 7 8 9 10
## $ un_factor : Factor w/ 2 levels "F1","F2": 1 2
## $ una_matriz: int [1:2, 1:3] 1 2 3 4 5 6
## $ un_df      : 'data.frame': 3 obs. of 2 variables:
## ..$ var_num : int [1:3] 0 1 2
## ..$ var_text: Factor w/ 3 levels "dos","tres","uno": 3 1 2
```

```
length(mi_lista)
```

```
## [1] 4
```

Estructuras de datos en R

Estructura de datos	Dimensiones	Tipos de datos mixtos	Posibilidad utilizar factores
vector	1	-	+
matrix	2	-	-
array	n	-	-
data.frame	2	+	+
list	1 (n)	+	+

Estructuras de datos en R

Ejercicio 2A

Elaborar una estructura de datos en R que corresponde a una ficha de registro de participación de un deportista en competencia (maratón). Escribir el código y visualizar el contenido de la estructura elaborada.

Ficha debe incluir:

- Nombre de deportista
 - Su nacionalidad
 - Su edad y género
 - Día de competencia
 - Coordenadas geográficas de los puntos de inicio, de mitad y de final del recorrido en maratón
 - Tiempo (horas, minutos, segundos por separado) cuando el deportista paso por cada uno de estos puntos
- | | |
|---------|-------------------|
| Inicio: | 19.3328, -99.1870 |
| Mitad: | 19.4407, -99.2046 |
| Final: | 19.4323, -99.1333 |

Extracción de datos en R

- Todas las estructuras de datos en R cuentan con índices de posición de elementos. Los índices son las secuencias numéricas de valores enteros consecutivos $1 \dots n$
- Los elementos en las estructuras de datos se puede extraer utilizando los índices mencionados.
- En caso de **vectores**, estos cuentan con la única dimensión, y acceso a sus elementos requiere el único índice.
- Los corchetes después de nombre del vector `[i]` se emplean para especificar el valor o valores del índice de elementos por consultar.

Ejemplos:

<code>c[3]</code>	– extraer el elemento 3 del vector “c”
<code>c[2:5]</code>	– extraer los elementos de 2 a 5 en del vector “c”
<code>c[c(1,4,6)]</code>	– extraer los elementos 1, 4 y 5 en del vector “c”

Extracción de datos en R

Ejemplo:
extracción desde vectores

```
meses <- c("enero", "febrero", "marzo", "abril",  
           "mayo", "junio", "julio", "agosto",  
           "septiembre", "octubre", "noviembre", "diciembre")
```

```
meses[3]
```

```
## [1] "marzo"
```

```
meses[2:5]
```

```
## [1] "febrero" "marzo"   "abril"   "mayo"
```

```
meses[5:2]
```

```
## [1] "mayo"   "abril"  "marzo"  "febrero"
```

```
meses[c(1,6,12)]
```

```
## [1] "enero"   "junio"   "diciembre"
```

```
meses.selectos <- meses[7:9]  
meses.selectos
```

```
## [1] "julio"   "agosto"  "septiembre"
```

```
meses[15]
```

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

Nota: los valores NA corresponden
a los elementos sin datos

Extracción de datos en R

- En caso de **matrices**, son las estructuras bidimensionales, y acceso a sus elementos requiere dos índices.
- Los corchetes después de nombre del matriz `[i, j]` se emplean para especificar los valores del índice líneas y columnas
- Resultados de extracción de matrices pueden ser vectores o matrices parciales

Ejemplos:

<code>m2[2, 3]</code>	– extraer el elemento de la segunda fila y tercera columna
<code>m2[1,]</code>	– extraer todos los elementos de la primera fila
<code>m2[, 2]</code>	– extraer todos los elementos de la segunda columna
<code>m2[c(1, 3), 2]</code>	– extraer el primer y tercer elemento de la segunda columna
<code>m2[c(1, 3), 1:2]</code>	– extraer la primera y tercera fila de la primera y segunda columna (resultado una matriz 2 por 2)

Extracción de datos en R

Ejemplo: extracción desde matrices

```
m2
```

```
##           [,1]      [,2]      [,3]  
## [1,] 2.410825 1.799516 6.4424895  
## [2,] 3.654689 3.187548 1.2440275  
## [3,] 8.489164 7.818309 0.2471724
```

```
m2[c(1,3),1:2]
```

```
##           [,1]      [,2]  
## [1,] 2.862807 9.148732  
## [2,] 2.473620 4.880261
```

```
m2[2,3]
```

```
## [1] 1.244027
```

```
m2[1,]
```

```
## [1] 2.410825 1.799516 6.442489
```

```
m2[,2]
```

```
## [1] 1.799516 3.187548 7.818309
```

```
m2[c(1,3),2]
```

```
## [1] 1.799516 7.818309
```

Extracción de datos en R

`lista[i]`

- Extracción de los elementos de una **lista** puede ser por medio de los índices numéricos similar a lo de vectores.
 - Extracción con este método tipo produce la lista.

`lista[[i]]`

- Para extraer el elemento como una estructura de datos asignada al elemento se requiere utilizar los corchetes dobles. Así se puede consultar un solo elemento, pero no grupos de elementos.

`lista["nombre"]`

- En lugar del índice numérico se puede utilizar los nombres de elementos en comillas dobles.

`lista[["nombre"]]`

- Un resultado similar a corchetes dobles se puede lograr utilizando el sintaxis con el símbolo \$ y el nombre del elemento de la lista después.

`lista$nombre`

- Cada uno de los elementos extraídos de la lista puede contar con propia estructura interna, que en su cuenta puede ser sujeta de extracción anidada.

Extracción de datos en R

Ejemplos de extracción de una **lista**:

`mi_lista[3]` – extraer el tercer elemento (como una lista)

`mi_lista[[3]]` – extraer el tercer elemento (como la estructura de datos propia del elemento)

`mi_lista["una_matriz"]` – extraer el tercer elemento por su nombre

`mi_lista[c("una_matriz", "un_df")]` – extraer dos elementos por nombre

`mi_lista$una_matriz` – forma alternativa de extraer elemento por nombre

`mi_lista$una_matriz[2,3]` – extracción anidada del elemento de matriz

Extracción de datos en R

Ejemplo: extracción desde una lista

```
mi_lista[3]
```

```
## $una_matriz
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
mi_lista[[3]]
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
mi_lista[c("una_matriz", "un_df")]
```

```
## $una_matriz
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
##
## $un_df
##   var_num var_text
## 1      0      uno
## 2      1      dos
## 3      2      tres
```

```
mi_lista["una_matriz"]
```

```
## $una_matriz
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
mi_lista$una_matriz
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
mi_lista$una_matriz[2,3]
```

```
## [1] 6
```

Extracción de datos en R

- Los **dataframes**, combinan características de matrices y listas.
- El acceso a los elementos de dataframe se puede realizar con índices numéricos y con nombres, o combinando dos enfoques.
- Resultados de extracción puede ser es otro dataframe o un vector.

`ciudades.df[2]`

`ciudades.df["nombre"]`

`ciudades.df[["nombre"]]`

`ciudades.df$nombre`

- La extracción con el único índice en corchetes simples `df[j]` regresa la columna o un conjunto de columnas en forma de un nuevo dataframe.
- En lugar del índice en corchetes se puede utilizar el nombre o nombres de las columnas.
- Acceso a valores en única columna puede ser con la notación de dobles corchetes o los nombres después del símbolo \$, similar a lo de listas.

Extracción de datos en R

```
ciudades.df[1,2]
```

```
ciudades.df[1:2,c(2,4)]
```

```
ciudades.df[,2]
```

```
ciudades.df[, "nombre"]
```

```
ciudades.df[1, ]
```

- La extracción con dos índices en corchetes `df[i, j]` permite acceder a un elemento o a grupo de elementos.
- La extracción omitiendo el primer índice en corchetes `df[, j]` permite extraer las columnas completas.
- En lugar de índices se puede utilizar nombres de columnas.
- La extracción omitiendo el segundo índice en corchetes `df[i,]` permite extraer las filas completas.

Extracción de datos en R

Ejemplos de extracción desde un **dataframe**:

- | | |
|--|---|
| <code>ciudades.df[2]</code> | – extraer la segunda columna (como un dataframe) |
| <code>ciudades.df[2:4]</code> | – extraer las columnas de dos a cuatro |
| <code>ciudades.df["nombre"]</code> | – extraer la columna por nombre (como un dataframe) |
| <code>ciudades.df\$nombre</code> | – extraer la columna por nombre (como un vector, solo se puede acceder a una columna) |
|
 | |
| <code>ciudades.df[2,]</code> | – extraer la fila dos (como un vector) |
| <code>ciudades.df[,2]</code> | – extraer la columna dos (como un vector) |
|
 | |
| <code>ciudades.df[,c("nombre", "poblacion")]</code> | – dos columnas |
| <code>ciudades.df[1,c("nombre", "poblacion")]</code> | – una fila, dos columnas |
| <code>ciudades.df[c(1,4),c(2,4)]</code> | – dos filas, dos columnas |

Extracción de datos en R

Ejemplo: extracción desde un dataframe

```
ciudades.df[2]
```

```
##              nombre
## 1 Ciudad de México (ZMCM)
## 2      Guadalajara (AMG)
## 3      Monterrey (ZMM)
## 4              León
```

```
ciudades.df[2:4]
```

```
##              nombre      tipo poblacion
## 1 Ciudad de México (ZMCM) megaciudad    20.40
## 2      Guadalajara (AMG)    ciudad      4.75
## 3      Monterrey (ZMM)    ciudad      4.69
## 4              León    ciudad      1.24
```

Extracción de datos en R

```
ciudades.df["nombre"]
```

```
##              nombre  
## 1 Ciudad de México (ZMCM)  
## 2      Guadalajara (AMG)  
## 3      Monterrey (ZMM)  
## 4              León
```

```
ciudades.df$nombre
```

```
## [1] "Ciudad de México (ZMCM)" "Guadalajara (AMG)"  
## [3] "Monterrey (ZMM)"         "León"
```

```
ciudades.df[,2]
```

```
## [1] "Ciudad de México (ZMCM)" "Guadalajara (AMG)"  
## [3] "Monterrey (ZMM)"         "León"
```

Extracción de datos en R

```
ciudades.df[2,]
```

```
##   id          nombre  tipo poblacion latitud longitud
## 2   2 Guadalajara (AMG) ciudad      4.75  20.667 -103.392
```

```
ciudades.df[,c("nombre", "poblacion")]
```

```
##              nombre poblacion
## 1 Ciudad de México (ZMCM)    20.40
## 2      Guadalajara (AMG)     4.75
## 3      Monterrey (ZMM)     4.69
## 4              León       1.24
```

```
ciudades.df[1,c("nombre", "poblacion")]
```

```
##              nombre poblacion
## 1 Ciudad de México (ZMCM)    20.4
```

Estructuras de datos en R

Ejercicio 2B

Considerando la ficha elaborada en el ejercicio 1. Consulta en su ficha:

- a) Nombre de deportista
- b) Momento de tiempo cuando deportista inicio el maratón
- c) Momento de tiempo cuando deportista concluyo el maratón
- d) * Calcula cuanto tiempo le llevo recorrer toda la distancia (puede ser en horas, minutos o segundos)
- e) ** Genera en R un texto que dice: *Participante nombre logro recorrer el maratón en HH horas, MM minutos, SS segundos.*

Estructuras de datos en R

Ejercicio 2B

Puedes realizar las operaciones aritméticas con valores numéricos.

Explora algunas las funciones que pueden ser necesarias para realizar los puntos d) y e) del ejercicio 2.

`unlist()` – convertir valores de lista o dataframe a un vector

`paste()` – concatenar cadenas de texto

`floor()` – extraer parte entera de un número

`striptime()` – convertir entre tiempos y cadenas de texto

`difftime()` – calcular diferencia entre dos momentos de tiempo

PROFACAD: Producto 3 para el portafolio

Producto para el portafolio: Producto 3. Ejercicios 2A y 2B

Fecha de entrega del producto: 21 de julio 2019

Actividades: En equipos de 2 alumnos definir en R la estructura de datos que corresponde a la ficha de registro de actividad de un maratonista, y realizar consulta de información guardada en la ficha (ejercicios 2A y 2B).

Producto para el portafolio: Reporte de generación de ficha y su consulta en el formato del documento R Markdown PDF con los fragmentos de código insertados, que cumple con las características definidos en las condiciones de los ejercicios 2A y 2B.

Estructuras de datos en R

Bibliografía complementaria

Mendoza Vega, J. B. (2018). *R para principiantes*. Libro electrónico recuperado de <https://bookdown.org/jboscomendoza/r-principiantes4/>

Peng, R. D. (2016). *R Programming for Data Science*. Leanpub. <https://leanpub.com/rprogramming>

Wickham. H. y Grolemond, G. (2017). *R for Data Science*. O'Reilly. <http://r4ds.had.co.nz/>

Respuesta a ejercicio 2

<https://github.com/vshalisko/R-intro-UdeG/tree/master/DataStructures>