

# Unificando vectores en R

## Sesión 4

Natalie Julian - [www.nataliejulian.com](http://www.nataliejulian.com)

Estadística UC y Data Scientist en Zippedi Inc.

# League of legends

Hoy trabajaremos con información de League of Legends, un videojuego online multijugador, de batalla y estrategia en equipo, desarrollado por Riot Games.

Un/a jugador/a de League of legends decidió realizar un experimento, probar varios personajes y así estudiar y analizar estadísticas de sus victorias 😊 y derrotas ☹

La información se encuentra en el archivo `League.RData` para cargar los datos, utilice la siguiente línea de código:

```
load(file.choose())
```

Se abrirá una ventana en la que usted debe seleccionar la ubicación del archivo `League`.

# Cargando un archivo RData

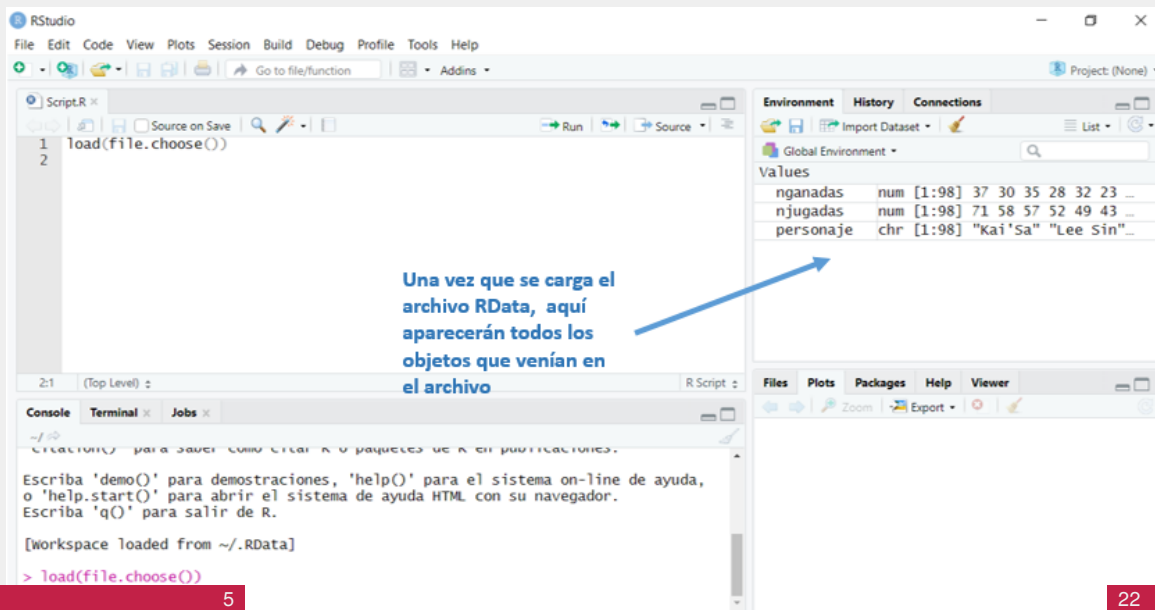
The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains the code `load(file.choose())`.
- Environment Panel:** Shows 'Global Environment' and 'Environment is empty'.
- File Selection Dialog:** Titled 'Seleccionar archivo', it displays a list of files. The file 'League' is selected. The dialog includes a search bar, a list of files with columns for 'Nombre' and 'Fecha de modificación', and buttons for 'Abrir' and 'Cancelar'.
- Console:** Displays the message `[Workspace loaded from ~/.RData]`.

Below the console output, the following R code is shown:

```
> load(file.choose())
```

# Cargando un archivo RData



Una vez que se carga el archivo RData, aquí aparecerán todos los objetos que venían en el archivo

Script.R x

```
1 load(file.choose())
2
```

Run Source

Environment History Connections

Global Environment

Values

Object	Type	Length	Values
nganadas	num	[1:98]	37 30 35 28 32 23 ...
njugadas	num	[1:98]	71 58 57 52 49 43 ...
personaje	chr	[1:98]	"Kai'Sa" "Lee Sin"...

Files Plots Packages Help Viewer

Console

```
~/... para saber como crear R o paquetes de R en publicaciones.
Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

[Workspace loaded from ~/.RData]
> load(file.choose())
```

5

22

# ¿Qué datos se han cargado en nuestra sesión?

Se han cargado los vectores `nganadas`, `njugadas` y `personaje`, donde:

- `personaje` indica el nombre del personaje
- `njugadas` cantidad de partidas jugadas con el personaje
- `nganadas` cantidad de partidas ganadas con el personaje

# Datos de League of legends

## nganadas

```
[1] 37 30 35 28 32 23 21 22 17 16 17 16 13 11 12 13 11 11 13 8 11 8 7 9 5
[26] 10 5 8 5 7 5 4 3 5 7 6 6 6 5 5 4 4 4 3 3 3 3 1 5 5
[51] 4 1 3 3 3 0 3 2 2 1 1 1 2 2 2 2 2 0 2 1 1 2 2 1 1
[76] 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## njugadas

```
[1] 71 58 57 52 49 43 38 36 35 35 33 32 29 25 24 23 22 21 20 19 18 18 18 17 16
[26] 15 15 14 14 13 13 13 12 11 10 10 9 9 9 9 9 8 8 8 8 8 8 8 7 7
[51] 7 7 6 6 6 6 5 5 5 5 5 5 4 4 4 4 4 4 3 3 3 2 2 2 2
[76] 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

## personaje

[1] "Kai'Sa"	"Lee Sin"	"Xayah"	"Nautilus"	"Gragas"
[6] "Rakan"	"Gangplank"	"Ryze"	"Akali"	"Thresh"
[11] "Renekton"	"Elise"	"Syndra"	"Ezreal"	"Kayle"
[16] "Qiyana"	"Rek'Sai"	"LeBlanc"	"Leona"	"Jayce"
[21] "Gnar"	"Tristana"	"Jarvan IV"	"Vladimir"	"Lucian"
[26] "Yasuo"	"Galio"	"Camille"	"Alistar"	"Orianna"
[31] "Yuumi"	"Pyke"	"Nocturne"	"Aatrox"	"Ornn"
[36] "Olaf"	"Taliyah"	"Varus"	"Kled"	"Morgana"
[41] "Twisted Fate"	"Cassiopeia"	"Corki"	"Blitzcrank"	"Garen"
[46] "Irelia"	"Shen"	"Skarner"	"Kennen"	"Tahm Kench"
[51] "Heimerdinger"	"Sion"	"Lissandra"	"Neeko"	"Veigar"
[56] "Mordekaiser"	"Rumble"	"Cho'Gath"	"Zoe"	"Azir"
[61] "Ekko"	"Karthus"	"Ashe"	"Fiora"	"Karma"
[66] "Quinn"	"Sivir"	"Caitlyn"	"Draven"	"Jinx"
[71] "Poppy"	"Fiddlesticks"	"Wayne"	"Hecarim"	"Jhin"
[76] "Kha'Zix"	"Lulu"	"Sejuani"	"Sylas"	"Jax"
[81] "Kog'Maw"	"Malphite"	"Pantheon"	"Urgot"	"Yorick"
[86] "Anivia"	"Bard"	"Braum"	"Lux"	"Nidalee"
[91] "Rengar"	"Sona"	"Vi"	"Viktor"	"Volibear"
[96] "Xin Zhao"	"Ziggs"	"Zilean"		



Supongamos que queremos obtener la información del número de partidas jugadas y la probabilidad de victoria unificada en una matriz.

# Unificando la información en R

Podemos unificar la información de varios vectores usando los comandos `cbind()`, `rbind()`, `matrix()` o `data.frame()`.

# `cbind()` y `rbind()`

Si utilizamos `cbind()`, la información se guardará verticalmente:

v	v	.....
e	e	.....
c	c	.....
t	t	.....
o	o	.....
r	r	.....
1	2	.....

Si utilizamos `rbind()`, la información se guardará horizontalmente:

vector1
vector2
.
.
.
.
.

# cbind()

```
datosvertical<-cbind(njugadas, probabilidad)
```

```
head(datosvertical, 7)
```

	njugadas	probabilidad
[1,]	71	0.5211268
[2,]	58	0.5172414
[3,]	57	0.6140351
[4,]	52	0.5384615
[5,]	49	0.6530612
[6,]	43	0.5348837
[7,]	38	0.5526316

```
dim(datosvertical)
```

```
[1] 98 2
```

# `rbind()`

```
datoshorizontal<-rbind(njugadas, probabilidad)
```

```
View(datoshorizontal)
```

```
dim(datoshorizontal)
```

```
[1]  2 98
```

```
class(datosvertical)
```

```
[1] "matrix"
```

```
class(datoshorizontal)
```

```
[1] "matrix"
```

# matrix()

```
matrix(c(njugadas, probabilidad), ncol=2)
dim(matrix(c(njugadas, probabilidad), ncol=2))
[1] 98  2
```

```
matrix(c(njugadas, probabilidad), nrow=98)
dim(matrix(c(njugadas, probabilidad), nrow=98))
[1] 98  2
```

```
class(matrix(c(njugadas, probabilidad), nrow=98))
[1] "matrix"
```

# ¿Qué pasa si queremos unificar toda la información de League of Legends?

Podríamos utilizar cbind pues tenemos la información estructurada de manera vertical:

```
leaguecbind<-cbind(personaje, njugadas, nganadas, probabilidad)
```

```
head(leaguecbind, 15)
```

	personaje	njugadas	nganadas	probabilidad
[1,]	"Kai'Sa"	"71"	"37"	"0.52112676056338"
[2,]	"Lee Sin"	"58"	"30"	"0.517241379310345"
[3,]	"Xayah"	"57"	"35"	"0.614035087719298"
[4,]	"Nautilus"	"52"	"28"	"0.538461538461538"
[5,]	"Gragas"	"49"	"32"	"0.653061224489796"
[6,]	"Rakan"	"43"	"23"	"0.534883720930233"
[7,]	"Gangplank"	"38"	"21"	"0.552631578947368"
[8,]	"Ryze"	"36"	"22"	"0.611111111111111"
[9,]	"Akali"	"35"	"17"	"0.485714285714286"
[10,]	"Thresh"	"35"	"16"	"0.457142857142857"
[11,]	"Renekton"	"33"	"17"	"0.515151515151515"
[12,]	"Elise"	"32"	"16"	"0.5"
[13,]	"Syndra"	"29"	"13"	"0.448275862068966"
[14,]	"Ezreal"	"25"	"11"	"0.44"
[15,]	"Kayle"	"24"	"12"	"0.5"

Tenemos problemas... ¿Teorías?

# Objetos `matrix`

Los objetos de tipo `matrix` asignan a todos sus elementos el mismo formato.



# Objetos `matrix`

Los objetos de tipo `matrix` operan de manera similar a los vectores, le asignan a todos sus elementos el mismo formato. Pero los datos no tienen por qué tener siempre el mismo formato...

# data.frame()

Podemos crear dataframes, objetos en los que se permite tener datos de distinto tipo.

```
df<-data.frame(personaje, njugadas, nganadas, probabilidad)
```

```
head(df, 10)
```

	personaje	njugadas	nganadas	probabilidad
1	Kai'Sa	71	37	0.5211268
2	Lee Sin	58	30	0.5172414
3	Xayah	57	35	0.6140351
4	Nautilus	52	28	0.5384615
5	Gragas	49	32	0.6530612
6	Rakan	43	23	0.5348837
7	Gangplank	38	21	0.5526316
8	Ryze	36	22	0.6111111
9	Akali	35	17	0.4857143
10	Thresh	35	16	0.4571429

```
str(df) #Indica cómo se lee cada variable
```

```
'data.frame': 98 obs. of 4 variables:
```

```
$ personaje : Factor w/ 98 levels "Aatrox","Akali",...: 34 43 91 52 25 64 22 69 2 80 ...  
$ njugadas : num 71 58 57 52 49 43 38 36 35 35 ...  
$ nganadas : num 37 30 35 28 32 23 21 22 17 16 ...  
$ probabilidad: num 0.521 0.517 0.614 0.538 0.653 ...
```

```
dim(df) #Dimensiones
```

```
[1] 98 4
```

```
nrow(df) #Número de filas
```

```
[1] 98
```

```
ncol(df) #Número de columnas
```

```
[1] 4
```

# Extraer elementos

## Extraer una variable completa

Por ejemplo, si quisiéramos extraer del objeto `df` la variable `personaje` utilizamos:

```
df$personaje #Entrega el vector de nombres
```

También, note que la variable `personaje` corresponde a la primera columna de `df`, por lo que, es posible también extraer dicha columna de la siguiente forma:

```
df[,1] #Entrega el vector de personajes
```

Análogamente se puede extraer la información de la variable `probabilidad` de las siguientes dos formas:

```
df$probabilidad # Entrega el vector de probabilidades de victoria  
df[,4] #Entrega el vector de probabilidades de victoria
```

## Extraer una fila completa

Por ejemplo, suponga que le interesa conocer la información del personaje 6, La información de este personaje se encuentra en la sexta fila del objeto `df`. Esto es posible utilizando la siguiente línea de código:

```
df[6,]
```

## Extraer una posición en particular

Si quisiéramos extraer solo la probabilidad de victoria para el tercer personaje se utiliza lo siguiente:

```
df[3,4]
```

## Extraer varias filas o columnas

Para extraer varias filas o columnas es análogo al procedimiento anterior, pero se debe especificar un vector. Por ejemplo, suponga que interesa conocer el nombre y probabilidad de victoria de los personajes 1, 3 y 5, esto se puede realizar de la siguiente forma:

```
df[seq(1, 5, by=2), c(1, 4)]
```

	personaje	probabilidad
1	Kai 'Sa	0.5211268
3	Xayah	0.6140351
5	Gragas	0.6530612

# Filtrar datos en una tabla de datos

Suponga que le interesa obtener la cantidad de partidas jugadas para todos aquellos personajes en los que la probabilidad de victoria es mayor a 0 y menor que 0.5. Utilizamos `which`:

```
df[which(df$probabilidad>0&df$probabilidad<0.5),]
```

# Funciones aplicables a dataframes

Función	Descripción
<code>dim()</code>	Dimensión de una planilla
<code>nrow()</code>	Número de filas (registros)
<code>ncol()</code>	Número de columnas (variables)
<code>names()</code>	Nombre de las columnas (variables)
<code>summary()</code>	Resumen estadístico de las variables
<code>head()</code>	Muestra los primeros registros de una planilla
<code>tail()</code>	Muestra los últimos registros de una planilla
<code>View()</code>	Muestra en una ventana aparte la planilla
<code>str()</code>	Muestra el tipo de variables que contiene la planilla
<code>class()</code>	Indica el formato de la planilla

*También es posible aplicar operaciones matemáticas (cuando la información es numérica), como las funciones que ya vimos, `log()`, `exp()`, `**`, `sqrt()`, `sin()`, entre otras.*