



RStudio creación de objetos

Octubre 28-29, 2019 Hermosillo, Sonora, México

Dr. Luis Alan Navarro Navarro

**Catedrático CONACyT-El Colegio de Sonora
Centro de Estudios en Gobierno y Asuntos
Públicos (CEGAP)**

alanphd.com

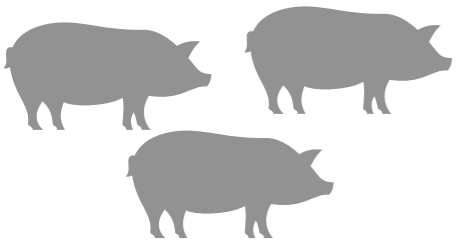
 @AlanNavarroPhd

¿Qué es un objeto?

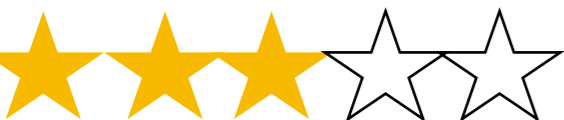
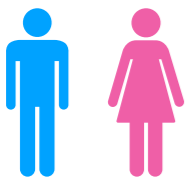
- *Desde el punto de vista de un usuario (no un programador), un objeto me sirve para almacenar datos.*
- *El objeto posee una estructura. Ésta me da las reglas de acceso a la información contenida en estos.*
- *El objeto es de un tipo o clase. Esto influye en la compatibilidad entre procesos.*
- *Algunos objetos los creo yo, en mis sesión; otros vienen dados en el paquete o programa. Alguno de estos objetos que vienen en el programa los puedo acceder y crear copias modificables.*
- *Outputs o resultados de análisis que normalmente son "texto y números" inertes en muchos programas, en R, pueden constituir un objeto.*

Vectores

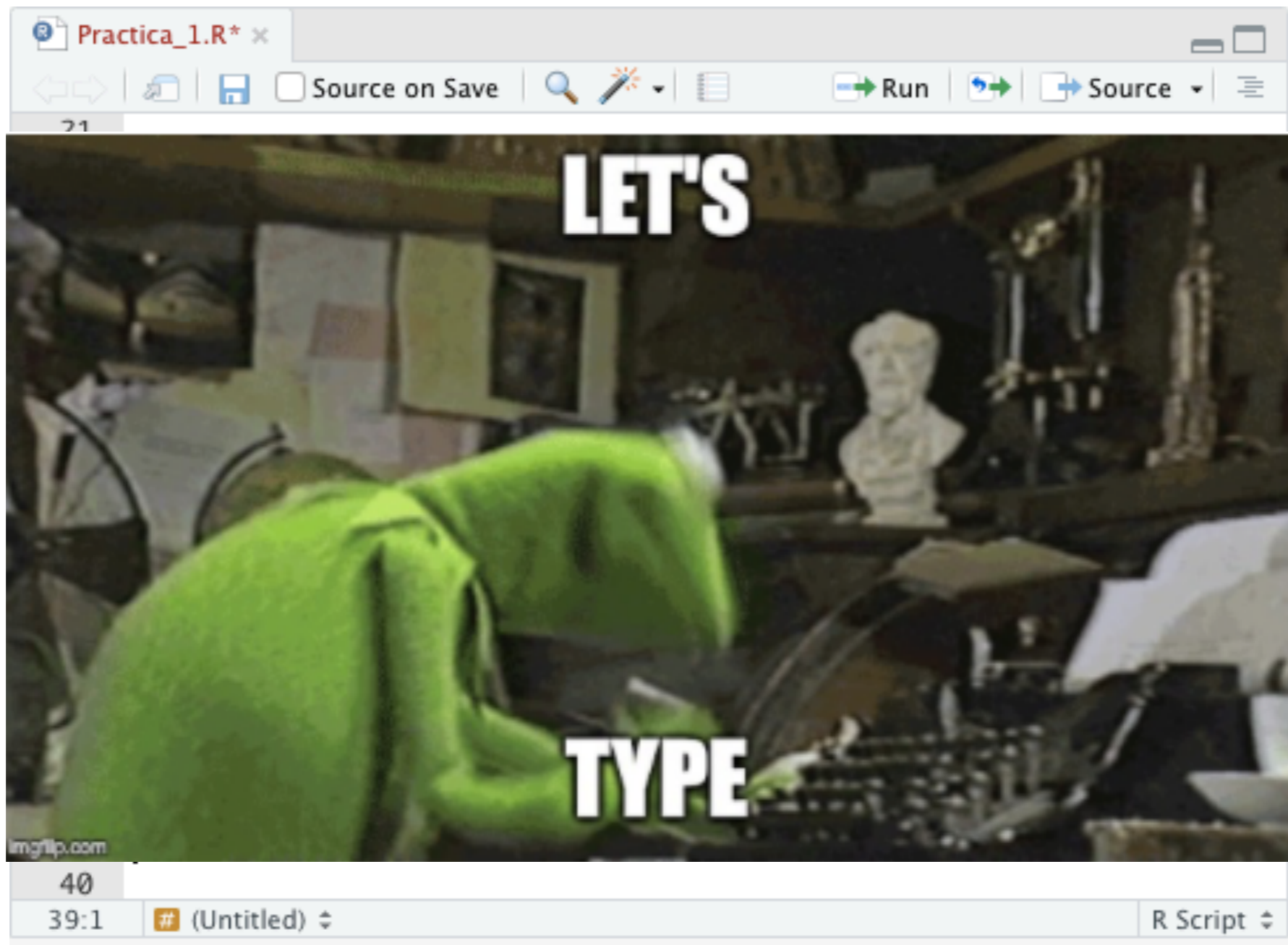
| Tipo | Variable | Ejemplo |
|----------|--------------------------------------|--|
| Numerico | Números con decimales | Por ejemplo: peso o altura de una persona |
| Entero | Números enteros | Cosas que no se pueden dividir, por ejemplo, número de empleados |
| Caracter | Letras del alfabeto | Por ejemplo: nombres de personas |
| Lógico | (TRUE, FALSE) | Comparaciones entre conjuntos |
| Factores | Variables categóricas ordenadas o no | Por ejemplo: Sexo, nivel de satisfacción, etc |



“pedro”, “rosa”



En RStudio en tu archivo de código:



En RStudio en tu archivo de código:

```
Practica_1.R* x
Source on Save
Run
Source

21
22 ## Sección: creamos un vector numérico
23 ## Fecha: Fri Oct 25 10:46:03 2019
24 #####
25 identificacion<- c(1:10) # c()
26 length(identificacion) # length()
27 class(identificacion) # class()
28
29 ## Sección: creamos un vector numérico al azar
30 ## Fecha: Fri Oct 25 10:46:53 2019
31 #####
32 set.seed(123) # set.seed()
33 edad<-round(runif(10, 20,26),0);edad # round (x,"numero de digitos")
34 # runif (n, nmin, nmax) genera 10 números aleatorios en el rango
35 # nmin al nmax
36 length(edad)
37 class(edad)
38
39 |
40
39:1 # (Untitled) R Script
```

En RStudio en tu archivo de código:

Usa archivo: Practica_1.R

Los nombres de los objetos "yo" los asigné de acuerdo a mis necesidades

```
21
22 ## Sección: creamos un vector numérico
23 ## Fecha: Fri Oct 25 10:46:03 2019
24 #####
25 identificacion<- c(1:10) # c()
26 length(identificacion) # length()
27 class(identificacion) # class()
28
29 ## Sección: creamos un vector numérico al azar
30 ## Fecha: Fri Oct 25 10:46:53 2019
31 #####
32 set.seed(123) # set.seed()
33 edad<-round(runif(10, 20,26),0);edad # round (x,"numero de digitos")
34 # runif (n, nmin, nmax) genera 10 números aletorios en el rango
35 # nmin al nmax
36 length(edad)
37 class(edad)
38
39 |
40
```

Una Snippet header_section

Estas son las funciones. Puedes siempre buscarlas en Internet.

39:1 # (Untitled) R Script

En RStudio en tu archivo de código:

```
edad <- round ( [redacted] , 0 )
```

redondeo a 0 dígitos.

este debe ser un objeto numérico
valor, vector, matriz,
u otra función.

```
runif (10, 20, 26)
```

genera 10 valores entre 20 y 26 (distribución uniforme)

Vectores

| ¿Qué? | Función |
|--|---------------------------------|
| Extensión del vector | <code>length(x)</code> |
| Tipo de elementos del vector | <code>class(x)</code> |
| Extraer el elemento "i" de un vector | <code>x[i]</code> |
| Extraer todos los elementos menos "i" | <code>x [-i]</code> |
| Extraer del elemento "i" al elemento "j" | <code>x [i:j]</code> |
| Extraer elementos específicos | <code>x [c(i, j, k)]</code> |



Extraer elementos de un vector:

```
38
39 ## Sección: extraer elementos de un vector
40 ## Fecha: Fri Oct 25 10:54:05 2019
41 #####
42 edad[1] # Elemento 1 del vector "edad"
43 edad[-2] # Todos menos el segundo elemento
44 edad[1:3] # Los primeros tres elementos
45 edad[c(5,7,10)] # Elementos 5, 7 y 10
46
47
```

45:38 # (Untitled) ↕ R Script ↕

Creamos un vector con elementos del tipo caracteres:

```
48 ## Sección: creamos variable categórica nominal
49 ## Fecha: Sat Oct 26 13:07:30 2019
50 #####
51 # crea una lista de mascotas
52 lista_mascotas <- c("perro", "gato", "pescadito", "pajarito", "tortuga", "hamster")
53 # Usamos esta funcion: sample(x, size, replace = FALSE, prob = NULL)
54 mascota <- sample(lista_mascotas, # de dónde tomo la muestra
55                   10, # tamaño de la muestra
56                   replace = TRUE, # con o sin reemplazo
57                   prob = c(0.4, 0.2, rep(0.1,4))) # probabilidad de cada categoría
58 mascota # llamamos el objeto para verlo en la consola
59 length(mascota)
60 class(mascota)
```

Creamos un vector con una variable categórica binaria:

```
63 ## Sección: creamos una variable categórica binaria
64 ## Fecha: Sat Oct 26 14:21:08 2019
65 #####
66 # 0 : Mujer; 1 : Hombre
67 genero <- c(1,0,1,1,1,0,0,0,1,1)
68 class(genero)
69 genero <- as.factor(genero) # as.factor()
70 levels(genero)=c("Mujer", "Hombre") # levels()
71 genero
72 class(genero)
73 # Para contar la frecuencia de cada categoría:
74 table(genero)
75
```

Creamos un vector con una variable categórica ordinal:

```
76 ## Sección: variable categórica ordinal
77 ## Fecha: Sat Oct 26 14:30:27 2019
78 #####
79 ##### Nivel de satisfaccion
80 satisf = c(3,1,2,2,1,3,1,2,3,2)
81 satisf <- factor(as.character(satisf), # factor()
82                 ordered = TRUE,
83                 levels = c("1", "2", "3"),
84                 labels= c("Bajo", "Medio", "Alto"))
85 satisf
86 class(satisf)
87 table(satisf)
88 ###
```

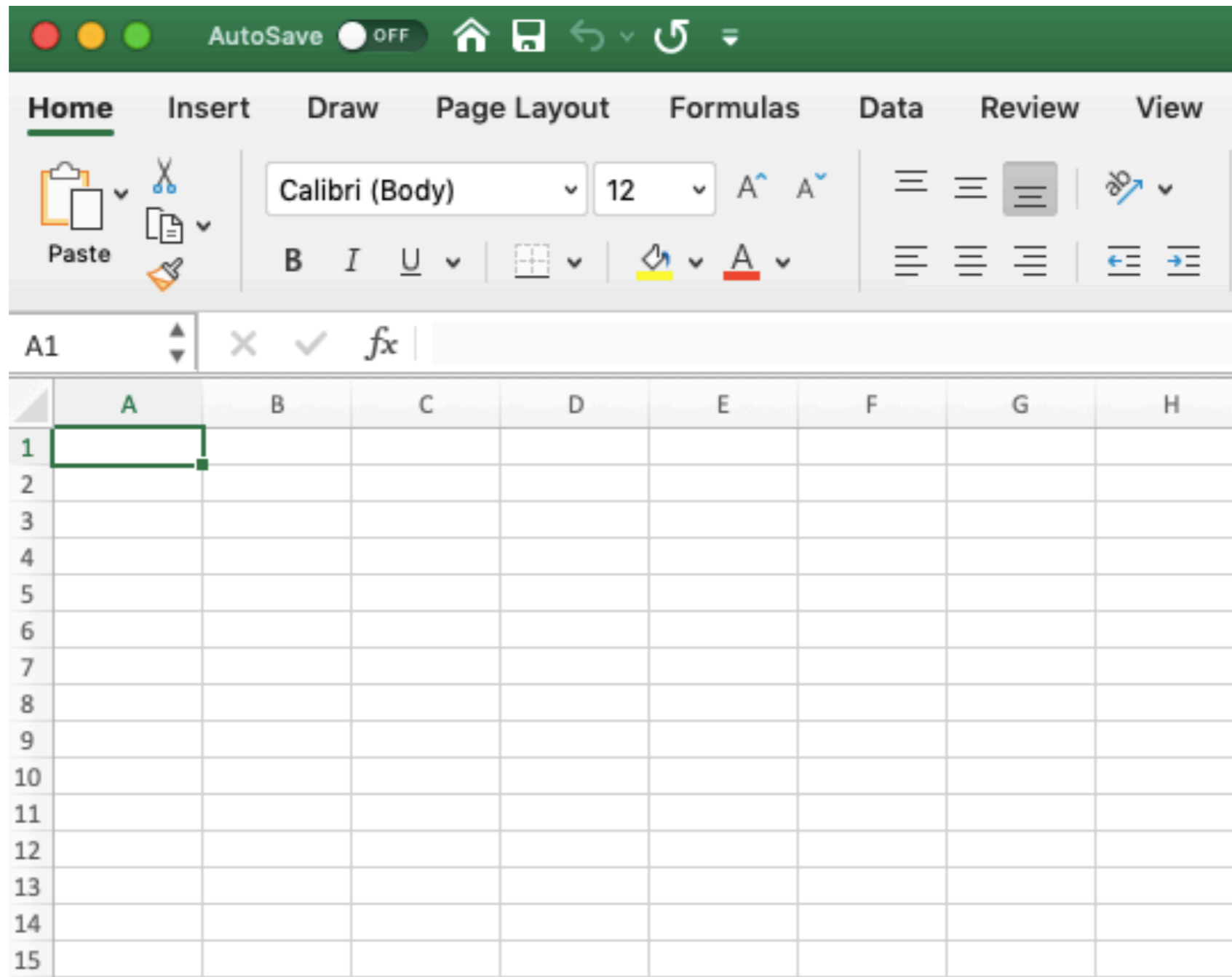
Relación entre dos variables (frecuencias) tabla 2x2:

```
90 ## Sección: relación entre variables
91 ## Fecha: Sat Oct 26 14:32:56 2019
92 - #####
93 table(genero,mascota)
94
95
```

Tabla o "dataframe"

| Obs | Variable_1 | Variable_2 | Variable_3 | Variable_4 | Variable_5 | Variable_6 |
|-----|------------|------------|------------|------------|------------|------------|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Tabla o "dataframe"



Es el típico formato de una hoja de Excel

Creamos un data frame con los vectores que ya tenemos:

```
97 ## Sección: creamos un data frame o una tabla
98 ## Fecha: Sat Oct 26 14:41:22 2019
99 - #####
100 df = data.frame(identificacion, edad, mascota, genero, satisf)
101 head(df,5) # las primeras 5 líneas del dataframe
102 tail(df,3) # las últimas 3 líneas del dataframe
103 View(df) # Visualizo la tabla
104 dim(df) # dimensiones en hileras y columnas
105 class(df) # tipo de objeto
```


El visor de dataframes:

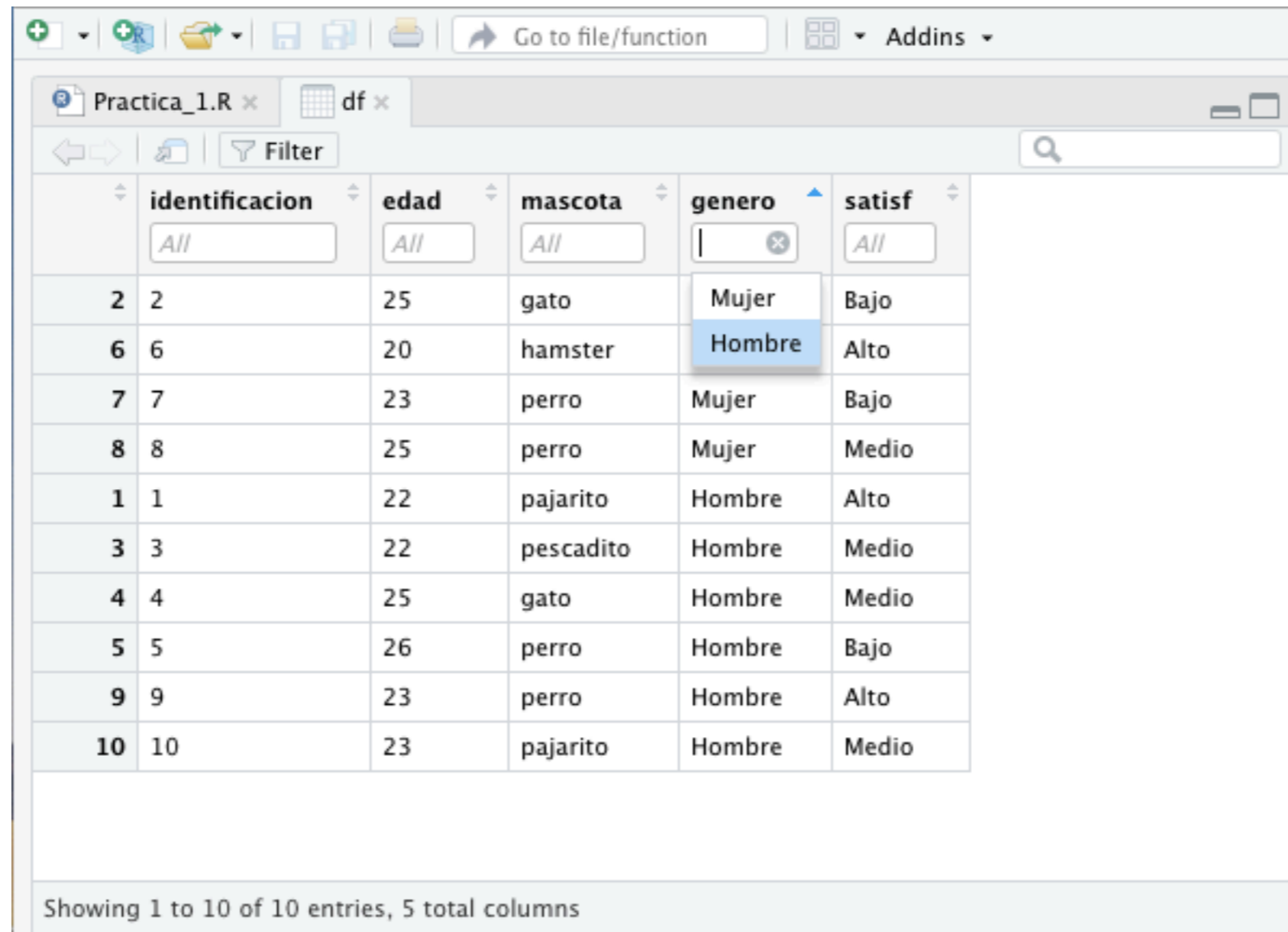
| | identificacion | edad | mascota | genero | satisf |
|----|----------------|------|-----------|--------|--------|
| 1 | 1 | 22 | pajarito | Hombre | Alto |
| 2 | 2 | 25 | gato | Mujer | Bajo |
| 3 | 3 | 22 | pescadito | Hombre | Medio |
| 4 | 4 | 25 | gato | Hombre | Medio |
| 5 | 5 | 26 | perro | Hombre | Bajo |
| 6 | 6 | 20 | hamster | Mujer | Alto |
| 7 | 7 | 23 | perro | Mujer | Bajo |
| 8 | 8 | 25 | perro | Mujer | Medio |
| 9 | 9 | 23 | perro | Hombre | Alto |
| 10 | 10 | 23 | pajarito | Hombre | Medio |

Showing 1 to 10 of 10 entries, 5 total columns

Aparece en el panel superior izquierdo como una pestaña en los archivos de código.

Es solo un visor.

El visor de dataframes:



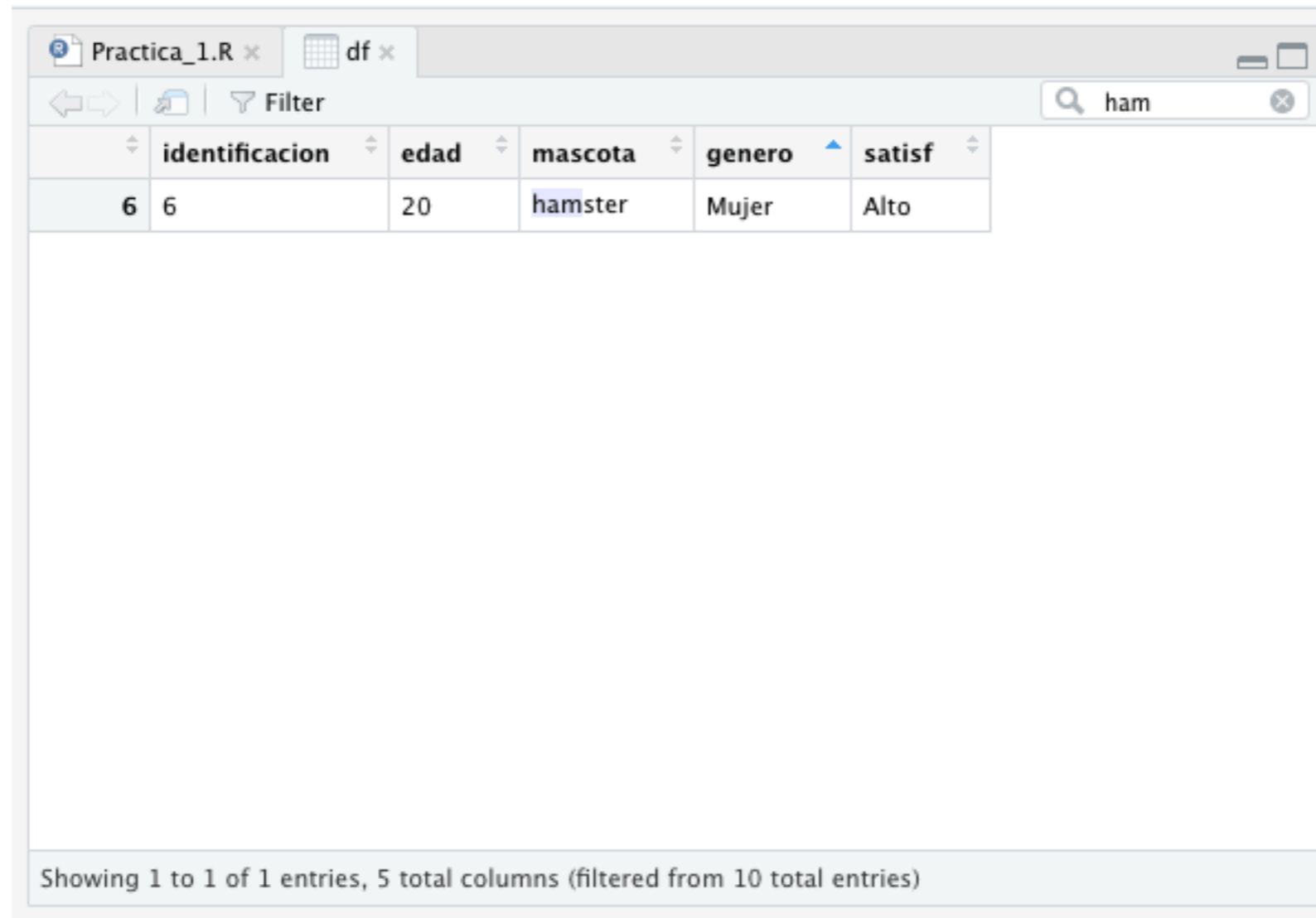
The screenshot shows a RStudio interface with a dataframe viewer. The dataframe has 10 rows and 5 columns: 'identificacion', 'edad', 'mascota', 'genero', and 'satisf'. The 'genero' column is currently filtered to show 'Mujer' and 'Hombre'. The 'satisf' column has values 'Bajo', 'Alto', and 'Medio'. The 'mascota' column has values 'gato', 'hamster', 'perro', and 'pajarito'. The 'edad' column has values 25, 20, 23, 25, 22, 22, 25, 26, 23, and 23. The 'identificacion' column has values 2, 6, 7, 8, 1, 3, 4, 5, 9, and 10.

| | identificacion | edad | mascota | genero | satisf |
|----|----------------|------|-----------|--------|--------|
| 2 | 2 | 25 | gato | Mujer | Bajo |
| 6 | 6 | 20 | hamster | Hombre | Alto |
| 7 | 7 | 23 | perro | Mujer | Bajo |
| 8 | 8 | 25 | perro | Mujer | Medio |
| 1 | 1 | 22 | pajarito | Hombre | Alto |
| 3 | 3 | 22 | pescadito | Hombre | Medio |
| 4 | 4 | 25 | gato | Hombre | Medio |
| 5 | 5 | 26 | perro | Hombre | Bajo |
| 9 | 9 | 23 | perro | Hombre | Alto |
| 10 | 10 | 23 | pajarito | Hombre | Medio |

Showing 1 to 10 of 10 entries, 5 total columns

Permite hacer filtro usando las variables.

El visor de dataframes:

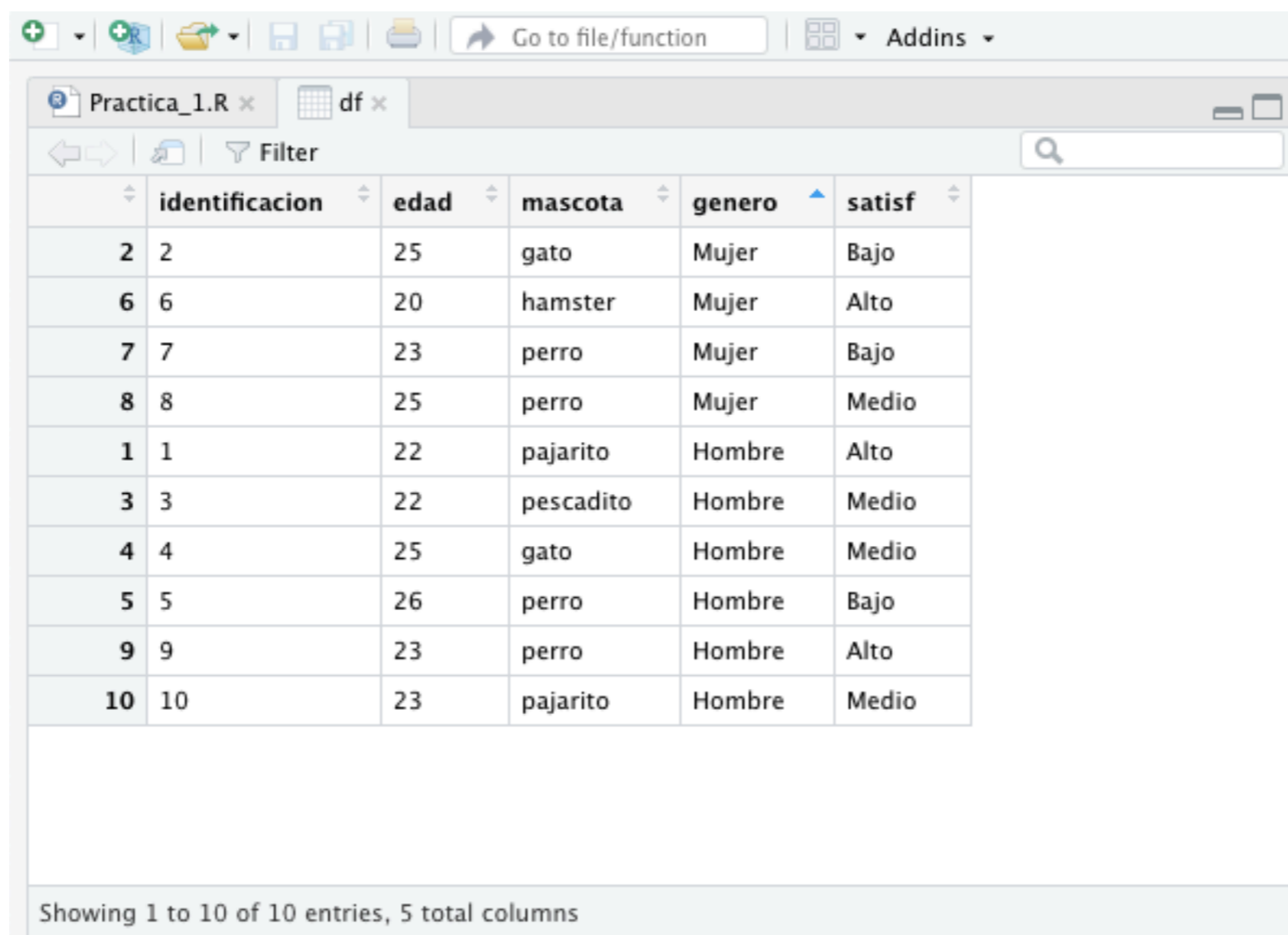


| | identificacion | edad | mascota | genero | satisf |
|---|----------------|------|---------|--------|--------|
| 6 | 6 | 20 | hamster | Mujer | Alto |

Showing 1 to 1 of 1 entries, 5 total columns (filtered from 10 total entries)

Hacer búsquedas.

El visor de dataframes:



Practica_1.R x df x

Filter

| | identificacion | edad | mascota | genero | satisf |
|----|----------------|------|-----------|--------|--------|
| 2 | 2 | 25 | gato | Mujer | Bajo |
| 6 | 6 | 20 | hamster | Mujer | Alto |
| 7 | 7 | 23 | perro | Mujer | Bajo |
| 8 | 8 | 25 | perro | Mujer | Medio |
| 1 | 1 | 22 | pajarito | Hombre | Alto |
| 3 | 3 | 22 | pescadito | Hombre | Medio |
| 4 | 4 | 25 | gato | Hombre | Medio |
| 5 | 5 | 26 | perro | Hombre | Bajo |
| 9 | 9 | 23 | perro | Hombre | Alto |
| 10 | 10 | 23 | pajarito | Hombre | Medio |

Showing 1 to 10 of 10 entries, 5 total columns

Ordenar por columnas.

Matrices

- *Arreglo rectangular de números. Cuadrada o rectangular. Dimensiones se expresa en hileras por columnas. Puede ser simétrica o asimétrica. En ocasiones la diagonal principal no está definida.*
- *¿Qué tipo de datos podemos almacenar en éstas?*
- *Comercio entre países. Acuerdos comerciales. Distancia entre ciudades. Países con frontera común o con un mismo idioma.*
- *Matrices de coeficientes técnicos, por ejemplo, insumo-producto.*
- *Sociomatrices que guardan las relaciones entre actores sociales, es decir, representan redes sociales.*
- *Una imagen (raster) se puede representar como una matriz.*

Matrices

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| a | | | | | | |
| b | | | | | | |
| c | | | | | | |
| d | | | | | | |
| e | | | | | | |
| f | | | | | | |

Tip

En realidad yo no escribo todo el código (from scratch), busco la función en Google, selecciono, copio, pego y lo adapto a mis necesidades.

It is possible to name the rows and columns of matrix during creation by passing a 2 element list to the argument `dimnames`.



```
> x <- matrix(1:9, nrow = 3, dimnames = list(c("X","Y","Z"), c("A","B","C")))
> x
  A B C
X 1 4 7
Y 2 5 8
Z 3 6 9
```

URL: <https://www.datamentor.io/r-programming/matrix/>

```
107
108 ## Sección: creamos una matriz
109 ## Fecha: Sat Oct 26 15:18:47 2019
110 - #####
111 x <- matrix(1:9, nrow = 3, dimnames = list(c("X", "Y", "Z"), c("A", "B", "C")))
112
113
114 |
```

Matrices

Matriz

Hileras x Columnas = Elementos

10 x 10 = 100

→ ocupa un vector de 100 elementos.

nrow = 10

→ número de hileras.

byrow = TRUE ó FALSE

→ si voy a llenar la matriz por hileras o por columnas

¿llevaron nombre las columnas e hileras?

dimnames =

→

Matrices

```
108 ## Sección: creamos una matriz
109 ## Fecha: Sat Oct 26 15:18:47 2019
110 - #####
111 x <- matrix(1:100,
112             nrow = 10,
113             dimnames = list(c(toupper(letters[1:10])), c(toupper(letters[1:10]))))
114 x
115 x <- matrix(1:100,
116             nrow = 10,
117             byrow = T,
118             dimnames = list(c(toupper(letters[1:10])), c(toupper(letters[1:10]))))
119 x
```

Así debe lucir tus paneles de RStudio:

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for creating a matrix and a data frame. The code includes comments in Spanish and uses `matrix()` and `df` functions.
- Environment Pane:** Shows the Global Environment with variables: `df` (data.frame), `edad` (numeric), `genero` (factor), `identificacion` (integer), `lista_mascotas` (character), `mascota` (character), `satisf` (ordered), and `x` (matrix).
- Console:** Shows the execution of the code, resulting in a 10x10 matrix of integers from 1 to 100, with columns labeled A through J.
- Files Pane:** Shows the file structure in the current directory, including `.RData`, `.Rhistory`, `Practica_1.R`, and `semana_i.Rproj`.

```
107
108 ## Sección: creamos una matriz
109 ## Fecha: Sat Oct 26 15:18:47 2019
110 #####
111 x <- matrix(1:100,
112             nrow = 10,
113             dimnames = list(c(toupper(letters[1:10])), c(toupper(letters[1:10]))))
114 x
115 x <- matrix(1:100,
116             nrow = 10,
117             byrow = T,
118             dimnames = list(c(toupper(letters[1:10])), c(toupper(letters[1:10]))))
119 x
120
121
122
123
124
125
126
127
```

```
> x <- matrix(1:100, nrow = 10, byrow = T, dimnames = list(c(toupper(letters[1:10])), c(toupper(letters[1:10]))))
> x
  A B C D E F G H I J
A 1 2 3 4 5 6 7 8 9 10
B 11 12 13 14 15 16 17 18 19 20
C 21 22 23 24 25 26 27 28 29 30
D 31 32 33 34 35 36 37 38 39 40
E 41 42 43 44 45 46 47 48 49 50
F 51 52 53 54 55 56 57 58 59 60
G 61 62 63 64 65 66 67 68 69 70
H 71 72 73 74 75 76 77 78 79 80
I 81 82 83 84 85 86 87 88 89 90
J 91 92 93 94 95 96 97 98 99 100
>
```

Muchas gracias