

Conceptos Básicos e Introducción a R

Curso de Muestreo Probabilístico en Encuestas de Hogares

Andrés Gutiérrez, Ph.D.

CEPAL - Unidad de Estadísticas Sociales

R y RStudio

Programación básica en R

Objetos: Factores y vectores

Objetos: archivos de datos

Librerías en R

Trabajo Práctico

R y RStudio

R

R es un software de acceso libre enfocado en análisis estadístico, minería de datos y visualización de información.

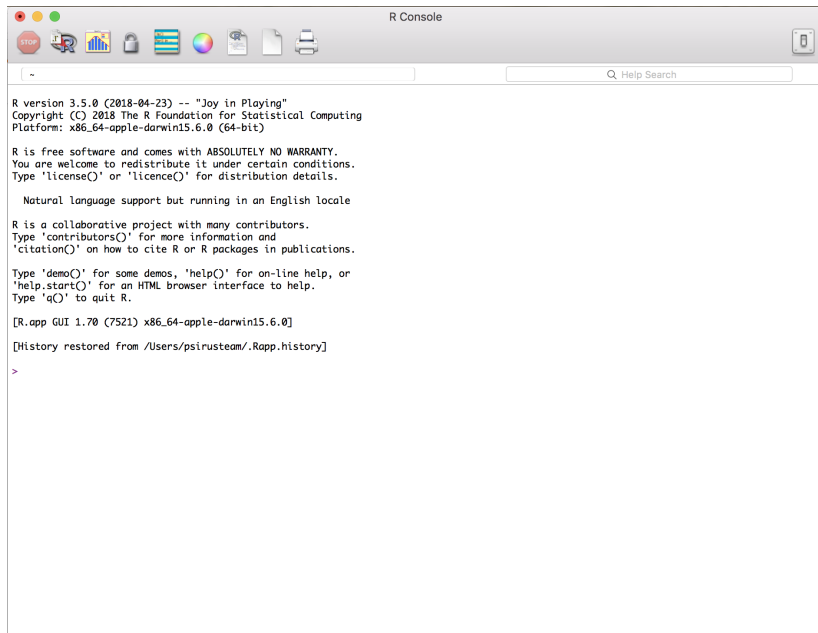
Para instalarlo en Windows:

1. Descargar la última versión del software (R-3.5.0) desde <https://cran.r-project.org/bin/windows/base/>
2. Abra el archivo ejecutable (.exe) y siga las instrucciones para instalar el software.

Para instalarlo en Mac:

1. Descargar la última versión del software (R-3.5.0) desde <https://cran.r-project.org/bin/macosx/>
2. Descomprima el software (.pkg) y siga las instrucciones de instalación.

Consola de R



```
R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.70 (7521) x86_64-apple-darwin15.6.0]
[History restored from /Users/psirusteam/.Rapp.history]

>
```

RStudio

RStudio es una interfaz amigable de acceso libre con muchas funcionalidades y facilidades para el usuario de R.

1. Descargar la última versión del software desde <https://www.rstudio.com/products/rstudio/download/>
2. Siga las instrucciones para instalar el software.
3. A partir de ahora utilice esta interfaz para usar R.

Consola de RStudio

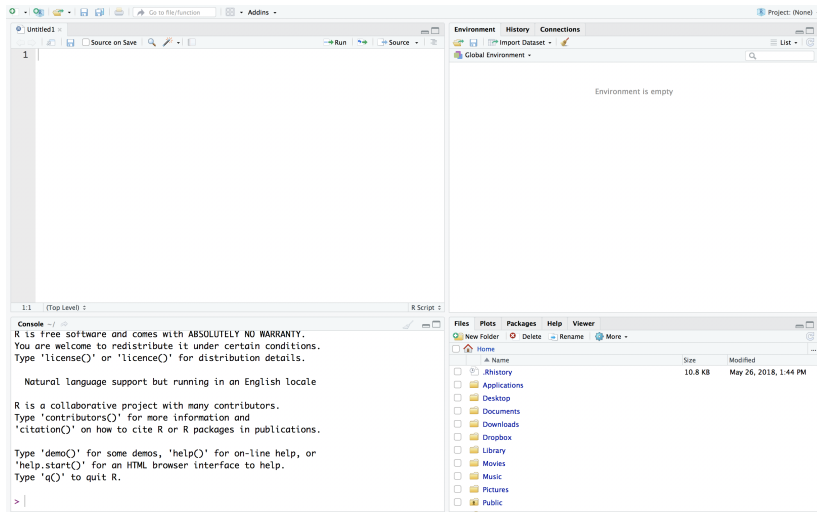


Figure 2: *Pantalla de RStudio*

Estructura de RStudio

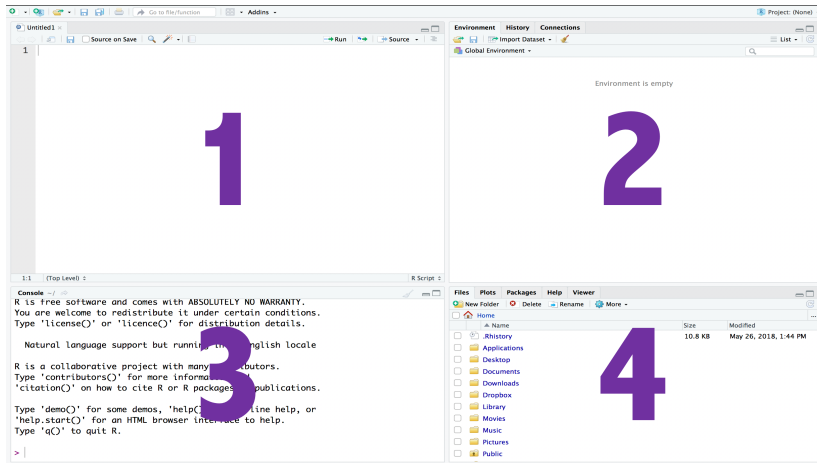


Figure 3: *Las cuatro ventanas dentro de RStudio*

Estructura de RStudio

La consola de RStudio está dividida en cuatro ventanas o paneles que se describen a continuación:

1. Panel de código: para escribir, editar y guardar los *scripts* (códigos computacionales) que son creados por el usuario.
2. Panel de objetos: en donde aparecen todos los objetos que son creados en la sesión junto con su estructura.
3. Consola de R: en donde se ejecuta directamente los comandos creados en el panel de código. También es posible hacer cálculos directamente en esta consola.
4. Panel auxiliar: en donde se visualizan los gráficos, las bases de datos, los directorios del sistema, así como la ayuda de las funciones de los paquetes.

Los beneficios de usar R

Algunas ventajas de utilizar este software son:

- ▶ Es gratis y de código abierto: cualquier persona puede modificar su código y mejorarlo.
- ▶ Multiplataforma: se ejecuta desde cualquier sistema operativo.
- ▶ Extensiones: una de las mayores ventajas de R es la posibilidad de extender sus funciones mediante la creación de librerías.
- ▶ Comunitario: existe una comunidad de colaboradores y usuarios que utilizan este software en la academia y en sus lugares de trabajo.
- ▶ Vínculo con otros lenguajes: R se conecta con otras herramientas computacionales como SAS, STATA, SPSS, ORACLE, SQL, entre muchos otros.

Los beneficios de usar R

- ▶ Es un lenguaje de programación creado por estadísticos para estadísticos.
- ▶ Es el lenguaje de programación estadística más usado en el mundo.
- ▶ Es soportado por *R Foundation for Statistical Computing*.
- ▶ Es un lenguaje de programación orientado a objetos, que en la notación estadística son representados por estructuras de datos.

The R-consortium

- ▶ El lenguaje R ha tenido un crecimiento significativo y ahora tiene más de dos millones de usuarios.
- ▶ Una amplia gama de industrias ha adoptado el lenguaje R, incluidas las industrias de biotecnología, finanzas, investigación y alta tecnología.
- ▶ El lenguaje R a menudo se integra en aplicaciones de análisis, visualización e informes de terceros.

The R-consortium

La misión central del *R-consortium* es trabajar y brindar apoyo a la *R Foundation for Statistical Computing* y a las organizaciones clave que desarrollan, mantienen, distribuyen y utilizan el software R a través de la identificación, el desarrollo y la implementación de proyectos de infraestructura.

The R-consortium

R Foundation Member



Platinum Members



Microsoft

GORDON AND BETTY
MOORE
FOUNDATION



Gold Members



Silver Members

alteryx



DataCamp



esri

Google



MANGO
SOLUTIONS

ORACLE®



PROCOGIA

Figure 4: *Miembros del consorcio*

Programación básica en R

Ejecución de códigos en RStudio

Para compilar el código desde RStudio siga las siguiente instrucciones.

1. Cree un nuevo archivo de R: **File - New File - R Script**.
2. Escriba los comandos en el panel de código.
3. Mantenga el cursor en la línea que desea compilar y ejecute la instrucción con los comandos **Ctrl + Enter** .

R como una calculadora

Considere los siguientes operadores aritméticos:

- ▶ Adición: +
- ▶ Sustracción: -
- ▶ Multiplicación: *
- ▶ División: /
- ▶ Potencia: ^
- ▶ Módulo: %%

R como una calculadora

R es un potente software que puede ser usado para realizar cálculos complejos. Empecemos con algunas operaciones sencillas.

```
3 + 5
```

```
## [1] 8
```

```
2 - 10
```

```
## [1] -8
```

```
2 * 3
```

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

```
2 / 5
```

```
## [1] 0.4
```

R como una calculadora

```
2 ^ 1
```

```
## [1] 2
```

```
2 ^ 10
```

```
## [1] 1024
```

```
5 %% 2
```

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

```
5 %% 5
```

```
## [1] 0
```

```
50 %% 3
```

```
## [1] 2
```

Operaciones lógicas

Las comparaciones lógicas en el entorno de R son las siguientes:

- ▶ Menor a: <
- ▶ Mayor a: >
- ▶ Menor o igual a: <=
- ▶ Mayor o igual a: >=
- ▶ Igual a: ==
- ▶ Diferente a: !=

Operaciones lógicas

Es muy importante tener un manejo adecuado de las funciones lógicas. A continuación algunos ejemplos.

```
3 > 5
```

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

```
3 < 4
```

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

```
2 == 2.01
```

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

```
3 >= 3
```

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

```
3 != 3
```

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

Funciones logarítmicas

```
log(2)
```

```
## [1] 0.6931472
```

```
log(2, base = 10)
```

```
## [1] 0.30103
```

```
log(2, 10)
```

```
## [1] 0.30103
```

```
log10(2)
```

```
## [1] 0.30103
```

Función exponencial

A continuación el manejo de la función exponencial.

```
exp(1)
```

```
## [1] 2.718282
```

```
exp(-2)
```

```
## [1] 0.1353353
```

```
exp(1000) ^ 10
```

```
## [1] Inf
```

Otras funciones útiles

Las variables categóricas en una encuesta de hogares se pueden ver como vectores de caracteres.

```
rep(3, 5)
```

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

```
rep("A", 5)
```

```
## [1] "A" "A" "A" "A" "A"
```

```
rep(NA, 10)
```

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

```
c(rep("A", 4), rep("B", 6))
```

```
## [1] "A" "A" "A" "A" "B" "B" "B" "B" "B" "B"
```


Objetos: Factores y vectores

Creación de objetos

R es un software que está enfocado en la programación orientada a objetos. En R todo es considerado un objeto que se define como una estructura de datos que tiene asociada unos atributos en particular.

La asignación de un valor a un objeto se realiza con los caracteres `<-` ó `=`.

```
Objeto <- valor
```

```
Objeto = valor
```

Creación de objetos

Note la diferencia entre un objeto de tipo numérico y un objeto de tipo caracter. Fíjese en que a medida que se ejecuta el código los objetos aparecen en el panel de objetos.

```
x <- 5; x
```

```
## [1] 5
```

```
is(x)
```

```
## [1] "numeric" "vector"
```

```
y <- "Hombre"; y
```

```
## [1] "Hombre"
```

```
is(y)
```

```
## [1] "character" "vector" "data.frame"
```

```
## [4] "SuperClassMethod"
```

Creación de vectores

Un vector es una arreglo de caracteres o números dentro de un objeto. Para crear un vector es necesario utilizar el comando `c()` y separar sus elementos con comas.

```
vector1 <- c(1, 2, 3, 4, 5)
vector1
```

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

```
vector2 <- c("Mi", "nombre", "es", "Andres")
vector2
```

```
## [1] "Mi"      "nombre" "es"      "Andres"
```

Operaciones con vectores

```
x1 <- c(1, 2, 3, 4)
x2 <- c(10, 10, 100, 1000)
5 * x1
```

```
## [1] 5 10 15 20
```

```
x1 + x2
```

```
## [1] 11 12 103 1004
```

```
x1 / x2
```

```
## [1] 0.100 0.200 0.030 0.004
```

Operaciones con vectores

```
sum(x1)
```

```
## [1] 10
```

```
sum(x2)
```

```
## [1] 1120
```

```
sum(x1) + sum(x2)
```

```
## [1] 1130
```

```
prod(x1) - prod(x2)
```

```
## [1] -9999976
```

Creación de objetos

Las variables continuas en una encuesta de hogares se pueden ver como vectores de números enteros o reales.

```
a <- c(3, 5, 10)
```

```
b <- c(2:20)
```

```
a
```

```
## [1] 3 5 10
```

```
b
```

```
## [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Funciones con objetos

```
a <- c(1:3)  
rep(a, 4)
```

```
## [1] 1 2 3 1 2 3 1 2 3 1 2 3
```

```
seq(1, 10, by = 1)
```

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

```
seq(0, 1, by = 0.1)
```

```
## [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```


Funciones con objetos

```
seq(0, 1, length.out = 13)
```

```
## [1] 0.00000000 0.08333333 0.16666667 0.25000000 0.33333333 0.41666667  
## [7] 0.50000000 0.58333333 0.66666667 0.75000000 0.83333333 0.91666667  
## [13] 1.00000000
```

```
d <- c(3, -1, 0, 4)  
sort(d)
```

```
## [1] -1 0 3 4
```

```
order(d)
```

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

Extrayendo componentes de los objetos

```
a <- c(-3:11)
```

```
a[4]
```

```
## [1] 0
```

```
a[-2]
```

```
## [1] -3 -1 0 1 2 3 4 5 6 7 8 9 10 11
```

```
a[c(1, 6, 7)]
```

```
## [1] -3 2 3
```

Extrayendo componentes de los objetos

```
a
```

```
## [1] -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11
```

```
a > 0
```

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

```
## [13] TRUE TRUE TRUE
```

```
a[a > 0]
```

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

Operadores Booleanos

- ▶ Disyunción: `|` * Una proposición es verdadera si alguno de sus componentes es verdadera.
- ▶ Conjunción: `&` * Una proposición es verdadera si y sólo si ambos componentes son verdaderas.

```
x <- c(1:10)
x[(x>8) | (x<5)]
```

```
## [1] 1 2 3 4 9 10
```

```
x[(x>8) & (x<5)]
```

```
## integer(0)
```

Disyunción

```
x > 8
```

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

```
x < 5
```

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

```
x > 8 | x < 5
```

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

```
x[c(T,T,T,T,F,F,F,F,T,T)]
```

```
## [1] 1 2 3 4 9 10
```

Conjunción

```
x > 8
```

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

```
x < 5
```

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

```
x > 8 & x < 5
```

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

```
x[c(F,F,F,F,F,F,F,F,F,F)]
```

```
## integer(0)
```

Creación de factores

Mediante la creación de factores se hacen explícitas las categorías de una variable discreta.

```
sexo <- c("Hombre", "Hombre", "Mujer",  
         "Mujer", "Hombre", "Mujer")  
sexo
```

```
## [1] "Hombre" "Hombre" "Mujer"  "Mujer"  "Hombre" "Mujer"
```

```
sexo1 <- as.factor(sexo)  
sexo1
```

```
## [1] Hombre Hombre Mujer  Mujer  Hombre Mujer  
## Levels: Hombre Mujer
```

Extrayendo componentes de los objetos

```
sexo1 == "Hombre"
```

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

```
which(sexo1 == "Hombre")
```

```
## [1] 1 2 5
```

```
H <- sexo1[sexo1 == "Hombre"]
```

```
M <- sexo1[sexo1 == "Mujer"]
```

```
H < M
```

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


La función summarize

Esta función permite resumir la información del factor mostrando una vista rápida de sus categorías y la cantidad de registros que tienen.

```
summary(sexo1)
```

```
## Hombre  Mujer  
##      3      3
```

```
summary(H)
```

```
## Hombre  Mujer  
##      3      0
```

```
summary(M)
```

```
## Hombre  Mujer  
##      0      3
```

Objetos: archivos de datos

Creación de un archivo de datos

Un archivo de datos (*data frame*) es un tipo de objeto que define una agrupación de variables que pueden ser numéricas, categóricas o mixtas.

- ▶ Las filas de un `data.frame` corresponden a los registros individuales de las unidades de observación.
- ▶ Las columnas de un `data.frame` corresponden a las variables medidas en esos individuos de interés.

Al analizar encuestas de hogares este tipo de objetos son los más apropiados para realizar procedimientos descriptivos e inferenciales.

Creación de las variables

A continuación crearemos algunas variables para la formación de nuestro primer archivo de datos.

```
# Identificación  
id <- paste0("CEPALIDk", c(1:5))  
# Otras variables de interés  
ingreso <- c(450, 500, 250, 1000, 500)  
gasto <- 100 + ingreso * 0.25  
zona <- c(rep("urbano", 3), rep("rural", 2))  
sexo <- c(rep("hombre", 2), rep("mujer", 3))
```

Creación de las variables

Utilizamos la función `data.frame` para crear el archivo de datos y sus propiedades.

```
datos <- data.frame(ID = id,  
                    INC = ingreso,  
                    EXP = gasto,  
                    ZON = zona,  
                    SEX = sexo)
```

Creación de las variables

Utilizamos la función `data.frame` para crear el archivo de datos y sus propiedades.

```
datos
```

ID	INC	EXP	ZON	SEX
CEPALIDk1	450	212.5	urbano	hombre
CEPALIDk2	500	225.0	urbano	hombre
CEPALIDk3	250	162.5	urbano	mujer
CEPALIDk4	1000	350.0	rural	mujer
CEPALIDk5	500	225.0	rural	mujer

Visualización de archivos de datos

En encuestas de hogares es muy común trabajar con bases de datos con miles o cientos de miles de registros. Para tener una visualización rápida y escueta de la base de datos se recomienda utilizar las siguientes funciones:

- ▶ `head()` imprime los primeros seis registros de la base.
- ▶ `tail()` imprime los últimos seis registros de la base.
- ▶ `names()` muestra los nombres de la base de datos.
- ▶ `str()` muestra la estructura de las variables de la base de datos.

Visualización de archivos de datos

```
head(datos)
```

ID	INC	EXP	ZON	SEX
CEPALIDk1	450	212.5	urbano	hombre
CEPALIDk2	500	225.0	urbano	hombre
CEPALIDk3	250	162.5	urbano	mujer
CEPALIDk4	1000	350.0	rural	mujer
CEPALIDk5	500	225.0	rural	mujer

Visualización de archivos de datos

```
names(datos)
```

```
## [1] "ID" "INC" "EXP" "ZON" "SEX"
```

```
str(datos)
```

```
## 'data.frame':    5 obs. of  5 variables:
## $ ID : Factor w/ 5 levels "CEPALIDk1","CEPALIDk2",...: 1 2 3 4 5
## $ INC: num  450 500 250 1000 500
## $ EXP: num  212 225 162 350 225
## $ ZON: Factor w/ 2 levels "rural","urbano": 2 2 2 1 1
## $ SEX: Factor w/ 2 levels "hombre","mujer": 1 1 2 2 2
```

Selección de casos

Se utilizan las llaves y la coma [,] para seleccionar casos particulares o variables particulares desde un conjunto de datos.

- ▶ Para seleccionar casos [casos,]
- ▶ Para seleccionar variables [, variables]
- ▶ Para seleccionar casos y variables [casos, variables]

Selección de casos

```
datos[1, ]
```

ID	INC	EXP	ZON	SEX
CEPALIDk1	450	212.5	urbano	hombre

```
datos[c(1, 4), ]
```

	ID	INC	EXP	ZON	SEX
1	CEPALIDk1	450	212.5	urbano	hombre
4	CEPALIDk4	1000	350.0	rural	mujer

Selección de variables

```
datos[, 3]
```

```
## [1] 212.5 225.0 162.5 350.0 225.0
```

```
datos[, c(2, 5)]
```

INC	SEX
450	hombre
500	hombre
250	mujer
1000	mujer
500	mujer

Selección de casos y variables

```
datos[1 , 1]
```

```
## [1] CEPALIDk1
```

```
## Levels: CEPALIDk1 CEPALIDk2 CEPALIDk3 CEPALIDk4 CEPALIDk5
```

```
datos[c(2, 5), c(2, 5)]
```

	INC	SEX
2	500	hombre
5	500	mujer

Trabajo con variables específicas

Se utiliza el operador \$ para trabajar con variables específicas. Para utilizar este enfoque es necesario conocer el rótulo de la variable.

```
datos$ID
```

```
## [1] CEPALIDk1 CEPALIDk2 CEPALIDk3 CEPALIDk4 CEPALIDk5
```

```
## Levels: CEPALIDk1 CEPALIDk2 CEPALIDk3 CEPALIDk4 CEPALIDk5
```

```
datos$EXP
```

```
## [1] 212.5 225.0 162.5 350.0 225.0
```

```
datos$SEX
```

```
## [1] hombre hombre mujer  mujer  mujer
```

```
## Levels: hombre mujer
```

Creación de nuevas variables

Habiendo definido un objeto de tipo `data.frame` es posible añadir tantas variables como sea requerido haciendo uso del operador `$`. Además es posible crear nuevas variables mediante la operación de otras variables.

```
datos$ESC <- c("Lee", "Lee", "Lee", "No lee", "Lee")  
datos$IOE <- datos$INC / datos$EXP
```

Creación de nuevas variables

datos

ID	INC	EXP	ZON	SEX	ESC	IOE
CEPALIDk1	450	212.5	urbano	hombre	Lee	2.117647
CEPALIDk2	500	225.0	urbano	hombre	Lee	2.222222
CEPALIDk3	250	162.5	urbano	mujer	Lee	1.538461
CEPALIDk4	1000	350.0	rural	mujer	No lee	2.857143
CEPALIDk5	500	225.0	rural	mujer	Lee	2.222222

Librerías en R

Librerías externas

Una librería de R es un conjunto de funciones y bases de datos encapsuladas en un objeto.

- ▶ Descarga de un paquete: `install.packages("paquete")`
- ▶ Uso de un paquete: `library("paquete")`
- ▶ Ayuda sobre un paquete: `help("paquete")`

TeachingSampling

La librería TeachingSampling fue desarrollada para seleccionar muestras probabilísticas bajo diferentes esquemas de muestreo en varias etapas. De la misma manera, puede ser usada para realizar los procesos apropiados de estimación de varianzas, entre otros.

```
install.packages("TeachingSampling")  
library("TeachingSampling")  
citation("TeachingSampling")
```

samplesize4surveys

La librería `samplesize4surveys` fue creada con el fin de permitirle al investigador evaluar rápidamente los escenarios apropiados para la determinación del tamaño de muestra en una encuesta de hogares.

```
install.packages("samplesize4surveys")  
library("samplesize4surveys")  
citation("samplesize4surveys")
```

Contenidos del paquete

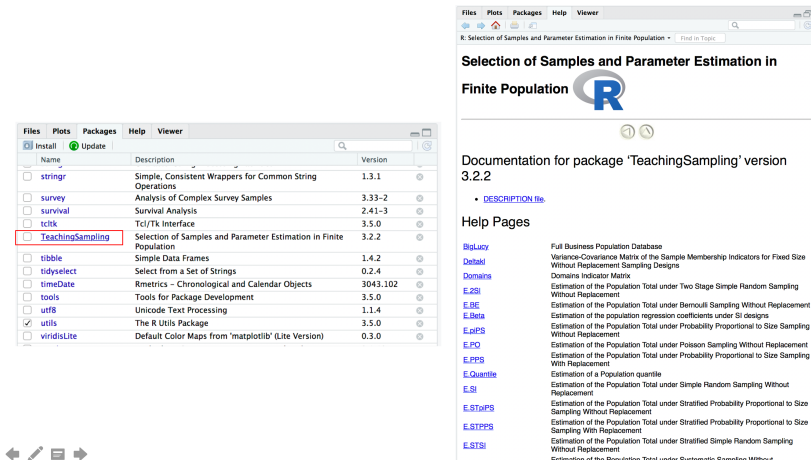
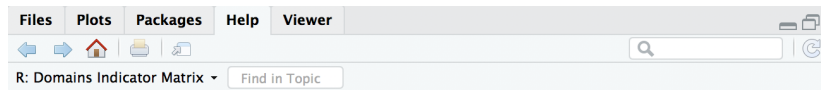


Figure 5: Acceso a la ayuda de una librería desde los paneles de RStudio

Ayuda de una función externa



Domains {TeachingSampling}

[R Documentation](#)

Domains Indicator Matrix

Description

Creates a matrix of domain indicator variables for every single unit in the selected sample or in the entire population

Usage

```
Domains(y)
```

Arguments

y Vector of the domain of interest containing the membership of each unit to a specified category of the domain

Details

Each value of **y** represents the domain which a specified unit belongs

Value

Usando una función externa

```
z <- c("Jefe", "Hijo", "Jefe", "Conyuge")  
Domains(z)
```

Conyuge	Hijo	Jefe
0	0	1
0	1	0
0	0	1
1	0	0

Accediendo a datos desde una librería

```
data(BigCity)
names(BigCity)
```

```
## [1] "HHID"      "PersonID"  "Stratum"   "PSU"       "Zone"
## [6] "Sex"       "Age"       "MaritalST" "Income"    "Expenditure"
## [11] "Employment" "Poverty"
```

```
is(BigCity$Income)
```

```
## [1] "numeric" "vector"
```


Visualización rápida

```
head(BigCity[,1:6])
```

HHID	PersonID	Stratum	PSU	Zone	Sex
idHH00001	idPer01	idStrt001	PSU0001	Rural	Male
idHH00001	idPer02	idStrt001	PSU0001	Rural	Female
idHH00001	idPer03	idStrt001	PSU0001	Rural	Female
idHH00001	idPer04	idStrt001	PSU0001	Rural	Male
idHH00001	idPer05	idStrt001	PSU0001	Rural	Male
idHH00002	idPer01	idStrt001	PSU0001	Rural	Male

```
head(BigCity[,7:12])
```

Age	MaritalST	Income	Expenditure	Employment	Poverty
38	Married	555.00	488.33	Employed	NotPoor
40	Married	555.00	488.33	Employed	NotPoor
20	Single	555.00	488.33	Inactive	NotPoor
19	Single	555.00	488.33	Employed	NotPoor
18	Single	555.00	488.33	Inactive	NotPoor
35	Married	298.34	216.70	Employed	Relative

Trabajo Práctico

Introducción a R

- Defina los siguientes vectores como objetos en R:

$$x_1 = (\log(6^3), \sqrt{365}, e^{-1}), x_2 = (\sin(2.8), \sqrt{8}, 36^{\frac{1}{3}}),$$

$$x_3 = (\text{Log}_{10}54, 0.6^{6+8.9}, \tan(3))$$

y calcule:

1. $x_1 + x_2$
2. $x_2 + x_3$
3. $3x_2 * x_1$
4. $\sqrt{x_1} + x_3$

Introducción a R

Utilizando la base de datos BigLucy que viene incluida en el paquete TeachingSampling realice los siguientes ejercicios (Para conocer la descripción de las variables puede utilizar ?BigLucy):

1. Encuentre la suma total de los impuestos de aquellas compañías que indicaron que utilizan internet o WEBmail para crear “self-propaganda”.
2. Cree una base de datos que contenga el Ingreso, el número de empleados y los impuestos de aquellas compañías que no están certificadas por la “International Organization for Standardization”.
3. De la base anterior, encuentre la media y mediana para el ingreso, número de empleados y los impuestos.
4. Compare los resultados anteriores con las medidas de la base completa.

Introducción a R

5. Cree una base de datos que contenga solo las compañías cuyos ingresos son mayores a 600, tengan un impuestos menor a 20 y que además tengan más de 25 años. ¿Cuántas compañías cumplen estas características?
6. Encuentre el Ingreso total de aquellas compañías que tienen más de 40 años o menos de 10 años en el mercado.

Introducción a R

Utilizando la libreria dplyr y la base de datos BigCity se crearán las bases de datos a nivel UPM y nivel hogar con las variables resumidas para cada una.

```
library(dplyr)

# Creación base de UPMS

BigCity$Ocupados = ifelse(
  BigCity$Employment == "Employed", 1, 0)
BigCity$Desocupados = ifelse(
  BigCity$Employment=="Unemployed", 1, 0)
BigCity$FT = BigCity$Ocupados + BigCity$Desocupados
```

Introducción a R

Utilizando la librería dplyr y la base de datos BigCity se crearán las bases de datos a nivel UPM y nivel hogar con las variables resumidas para cada una.

```
UPMS = BigCity %>% group_by(PSU) %>%  
  summarise(Hogares = length(unique(HHID)), # Nro de hogares en la UPM  
            Personas = n(),                # Nro de personas en la UPM  
            estratos = unique(Stratum),     # Estrato a la que pertenece la UPM  
            area = unique(Zone),            # Area de la UPM  
            hombres = mean(Sex == "Male"), # Proporción de hombres en la UPM  
            edad.media = mean(Age),         # Media edad de la UPM  
            ingreso.medio = mean(Income),   # Media ingreso de la UPM  
            gasto.medio = mean(Expenditure), # Media gasto de la UPM  
            Desempleo = sum(Desocupados)/sum(FT)) %>% # Tasa desempleo de la UPM  
as.data.frame()
```

Introducción a R

```
head(UPMS)
```

PSU	Hogares	Personas	estratos	area	hombres	edad.media
PSU0001	26	118	idStrt001	Rural	0.5254237	27.16949
PSU0002	32	136	idStrt001	Rural	0.4705882	26.39706
PSU0003	24	96	idStrt001	Rural	0.4791667	32.60417
PSU0004	22	88	idStrt001	Rural	0.4090909	29.59091
PSU0005	28	110	idStrt001	Rural	0.5090909	39.29091
PSU0006	30	116	idStrt001	Rural	0.4655172	29.48276

Introducción a R

```
# Creación base de hogares
```

```
Hogares = BigCity %>% group_by(HHID) %>%  
  summarise(Personas = n(),  
            estratos = unique(Stratum),  
            upm = unique(PSU),  
            area = unique(Zone),  
            hombres = mean(Sex == "Male"),  
            edad.media = mean(Age),  
            ingreso.medio = mean(Income),  
            gasto.medio = mean(Expenditure),  
            Desempleo = sum(Desocupados)/sum(FT),  
            pobreza = unique(Poverty)) %>%  
  as.data.frame()
```

Introducción a R

```
head(Hogares)
```

HHID	Personas	estratos	upm	area	hombres	edad.media
idHH00001	5	idStrt001	PSU0001	Rural	0.6	27.00
idHH00002	5	idStrt001	PSU0001	Rural	0.4	19.40
idHH00003	4	idStrt001	PSU0001	Rural	0.5	38.25
idHH00004	4	idStrt001	PSU0001	Rural	0.5	29.75
idHH00005	5	idStrt001	PSU0001	Rural	0.6	32.40
idHH00006	5	idStrt001	PSU0001	Rural	0.4	24.80

¡Gracias!

Andrés Gutiérrez

Experto Regional en Estadísticas Sociales

Division de Estadísticas

Email: andres.GUTIERREZ@cepal.org