Introducción al software estadístico

Módulo I

Nicolás Schmidt

nschmidt@cienciassociales.edu.uy

Departamento de Ciencia Política
Facultad de Ciencias Sociales
Universidad de la República

Estructura de la presentación

1. ¿Qué es R?
   - ¿Por qué usar R?
   - ¿Es útil programar?
   - CRAN
2. Instalación y estructura
   - Instalación
   - R
   - RStudio
   - Versiones
3. Estilos de programación
   - Script
   - Guías
4. Paquetes
   - Instalación
   - Paquetes pre-instalados y ruta de búsqueda
   - Recursos
5. Ayuda en R
   - Interna
   - Externa
6. Iniciar una sesión en R
   - Algunos comandos básicos
   - `options()`
   - Errores y advertencias en R
   - Uso de funciones

Estructura de la presentación

R

- R es un lenguaje de programación interpretado. No es compilado, se ejecuta paso a paso por un programa que interpreta comandos.
- R es un lenguaje de programación orientado al análisis estadístico y a la visualización gráfica de datos.
- La fecha de inicio de R es 1993 (aunque sus antecedentes se remontan a finales de los setenta con la creación del lenguaje S por parte de John Chambers y colaboradores).
- La versión 1.0.0 se liberó recién en febrero de 2000.
- Los nombres importantes de esta historia son muchos, pero los principales son: Robert Gentleman y Ross Ihaka. Ihaka relata una breve historia aquí y aquí
- R se distribuye con licencia GNU, el nombre importante de esta parte de la historia es: Martin Mächler.

¿Por qué usar R?

Algunas pocas razones...

1. Es libre y gratuito!
2. Funciona en múltiples plataformas (Windows, MAC, Linux)
3. Replicabilidad
4. Actualización de ultimas técnicas estadísticas
5. Está en constante desarrollo.
6. Potente herramienta de creación de gráficos.
7. Integración con otros lenguajes: LaTeX , C++, Python, Julia, Stan...

¿Por qué usar R?

Una razón fundamental: Cuenta con una comunidad muy activa.

**"When you talk about choosing programming languages,
I always say you shouldn't pick them based on technical merits,
but rather pick them based on the community."**

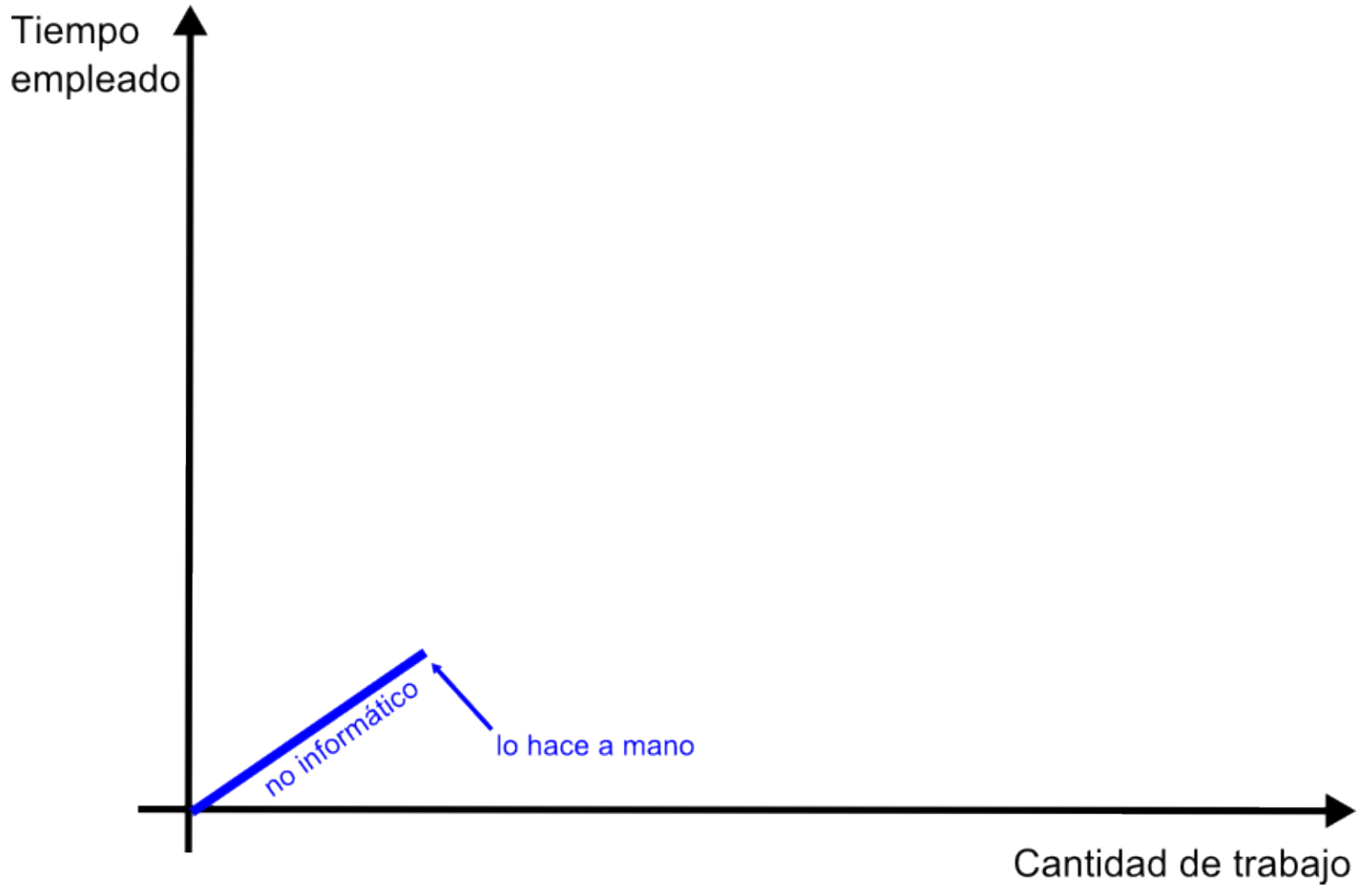
Hadley Wickham

Comunidad de usuarios de R en Uruguay

En Uruguay hay dos grupos:

- R-Ladies Montevideo $\Rightarrow$ Twitter: @RLadiesMVD
- Grupo de Usuarios de R::Montevideo $\Rightarrow$ Twitter: @gurumvd

¿Es útil programar?



Fuente: http://www.mclibre.org

Información sobre R y el CRAN

En la pagina de R (https://cran.r-project.org) se encuentra toda la información necesaria para iniciarse en R:

- Manuales principales aquí
- CRAN Repository Policy aquí
- Libros relacionados con S y R aquí
- Libros relacionados con S y R en distintos idiomas aquí
- R Journal aquí

CRAN (The Comprehensive R Archive Network) es una red ftp (File Transfer Protocol) que contiene toda la documentación sobre R y los paquetes.

Otros sitios donde hay información relevante sobre R:

- Journal of Statistical Software aquí
- Tendencias actuales en GitHub aquí
- R–blogger aquí

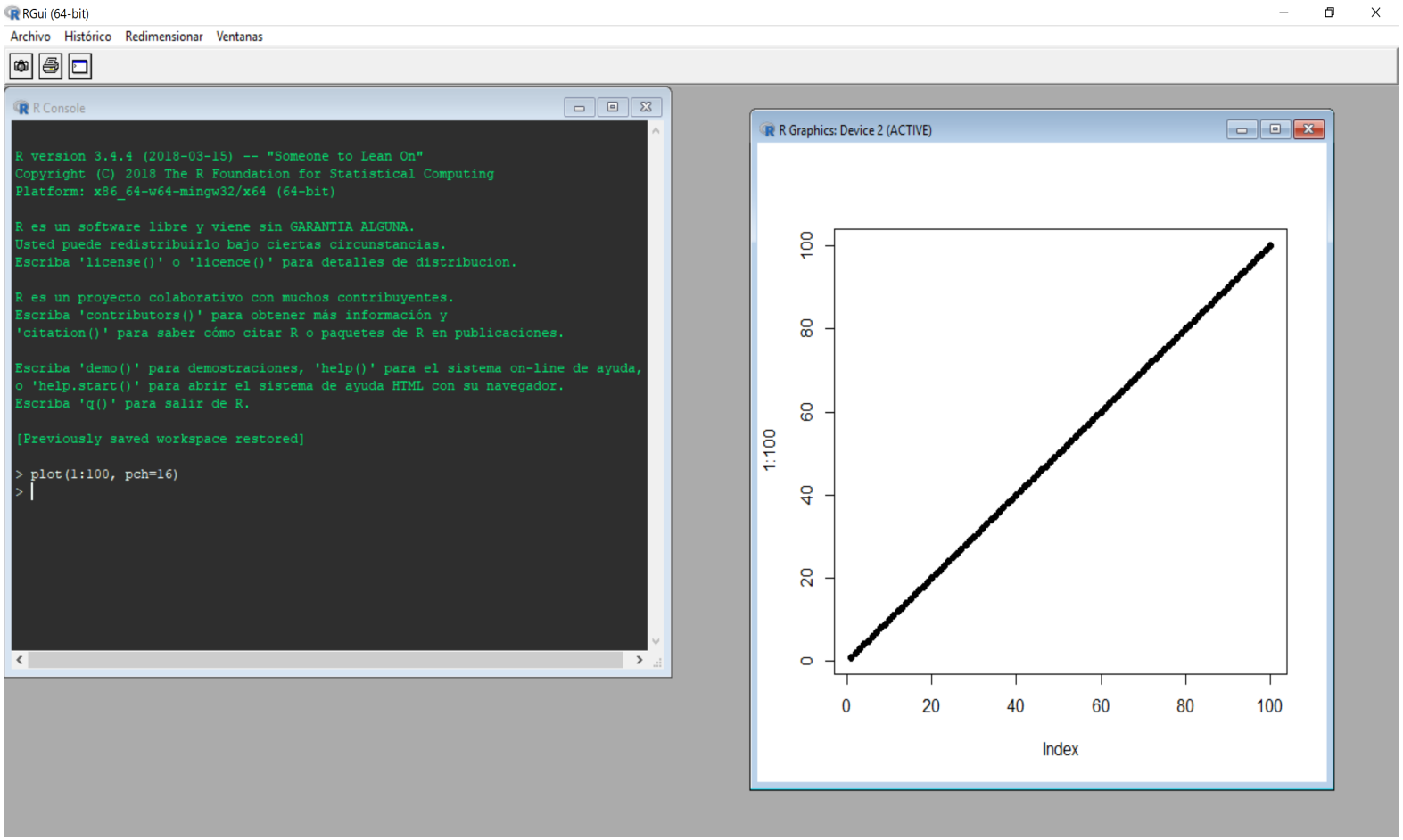
Estructura de la Presentación

Descarga, actualización y R en línea

1. Descarga
   - La descarga e instalación de R se hace desde este link: https://cran.r-project.org/
   - La descarga e instalación de RStudio se hace desde este link: https://www.rstudio.com
2. Actualización de R
   - Una manera simple:

```
install.packages("installr")
installr::updateR()
```

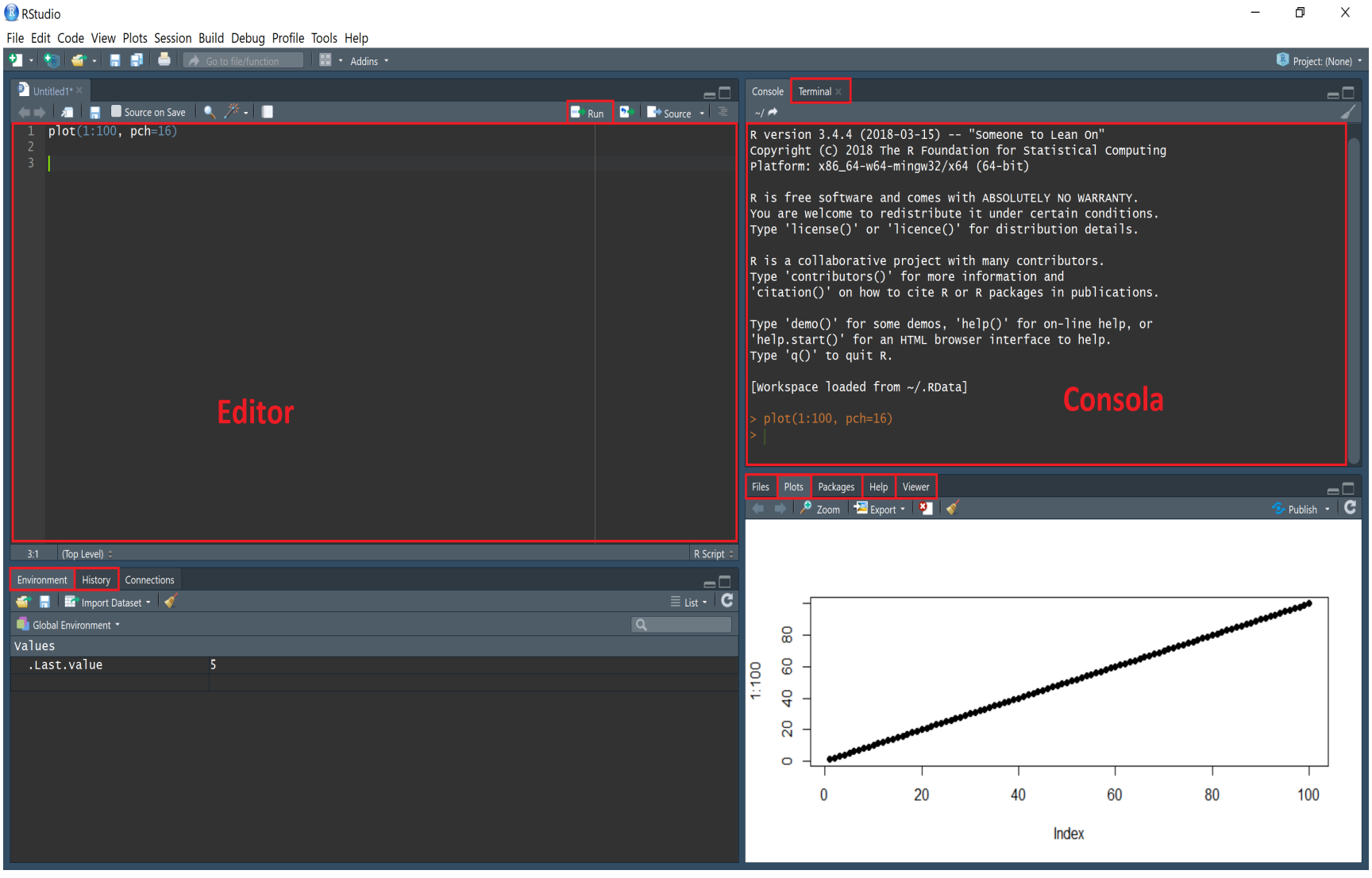
3. R Online
   - Es un R que se ejecuta lento, con poco procesamiento y sin poder cargar paquetes: http://rextester.com


RGui (64-bit)
Archivo Histórico Redimensionar Ventanas
R Console
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.
R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar 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.
[Previously saved workspace restored]
> plot(1:100, pch=16)
>
R Graphics: Device 2 (ACTIVE)
1:100
0
20
40
60
80
100
Index

Mensaje de inicio de R

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar 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.

[Previously saved workspace restored]

>
```

RStudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 plot(1:100, pch=16)
2
3
Editor
Console
Terminal
~/
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.
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.
[Workspace loaded from ~/.RData]
> plot(1:100, pch=16)
>
Consola
3:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Values
.Last.value
5
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
1:100
0
20
40
60
80
100
Index

Editor	Lugar para desarrollar scripts.
Consola	Imprime en pantallas las sentencias corridas desde el editor o directamente en consola
Run	Ejecuta una sentencia del editor (Ctrl+Enter) donde esté el cursor o lo que se seleccione
Environment	Lista todos los objetos generados en cada sesión
History	Historial de ejecuciones
Terminal	Un terminal para conectar con Git por ejemplo entre otras cosas
Files	Los archivos (datos, script) se pueden buscar desde esta pestaña
Plots	Imprime los gráficos
Packages	Es un gestor de los paquetes instalados
Help	Ayuda rápida

RStudio

Instrucciones y atajos sobre configuración y uso

→ Aquí encontraran algunas instrucciones sobre cómo customizar RStudio

→ Aquí encontraran algunas instrucciones sobre cómo usar el historial de comandos en RStudio.

→ Aquí encontraran algunos atajos para trabajar en consola.

Chequeo de versiones

Al iniciar R, en consola o en RStudio aparece la versión que se está ejecutando y el nombre.

Funciones útiles:

```
R.Version()                 # Información detallada de versión de R
R.Version()$version.string  # Número y fecha de versión
R.Version()$nickname        # Nombre de versión
sessionInfo()               # Información detallada de la sesión actual
RStudio.Version()           # Versión de RStudio
Sys.info()                  # Información del sistema operativo
```

Ejemplo:

```
R.Version()$version.string

## [1] "R version 3.4.1 (2017-06-30)"

R.Version()$nickname

## [1] "Single Candle"
```

Estructura de la Presentación

Importancia del uso de Scripts

Un script es un archivo en el que se escriben los comandos y las funciones que se desean guardar de una sesión de trabajo.

Confección: crear estos archivos es importante para tener un historial de trabajo.

Guardado: El nombre del archivo debe ser lo suficientemente claro como para reflejar el contenido y encontrarlo con facilidad. En lo posible debe ser corto y evitar el uso de ".".

Documentación: Es muy importante comentar las operaciones o creación de funciones que se realicen para poder replicar con facilidad el trabajo. Los comentarios deben iniciar con "#"

```
X <- c(1, 8, 19)
# El objeto 'X' es un vector de tres números. Y esto es un comentario.
```

Estilos de programación

Hay distintas maneras de programar. Es importante ser cuidadoso en la manera en la que se escribe código para que sea fácil replicar y leer el código.

Ejemplo:

```
# Mal
media<-function (x){n<-length(x);sum(x)/n}

# Bien
media <- function(x){
          n <- length(x)
          sum(x) / n
    }
```

- El paquete `styler` puede ser de gran ayuda!
- Guía de estilo de Hadley Wickham aquí
- Guía de estilo de Google aquí

Estructura de la Presentación

Un paquete de R es un conjunto de funciones que pertenecen a un mismo ambiente y que –por lo común– tienen una estructura general que justifica que estén juntas.

Descarga:

```
# Descargar un paquete desde CRAN
install.packages("esaps")
# la función 'install.packages("...")' descarga el paquete a la computadora
# (funciones y documentación de las mismas)

# Descargar un paquete desde GitHub (versión del paquete en desarrollo)
install.packages("devtools")
devtools::install_github("Nicolas-Schmidt/esaps")
```

Uso (cargar un paquete):

```
# Para usar las funciones de un paquete ya descargado hay que usar la función
# 'library(...)'
library("esaps")
# se puede prescindir del uso de las comillas.
# En el caso de la descarga el paquete debe estar entrecomillado
library(esaps)
```

Funciones sobre los paquetes

```r
installed.packages()    # lista paquetes instalados
packageStatus()         # informa cuántos paquetes no están actualizados
download.packages()     # compara versiones de los paquetes instalados
                        # con la actual en CRAN
available.packages()    # lista todos los paquetes disponibles
update.packages()       # actualiza paquetes
library(help="MASS")    # Información sobre un paquete específico
demo("graphics")        # Demostración de un paquete
```

Desde la pestaña 'Packages' de RStudio se pueden realizar la mayoría de estas funciones.

Cuando se carga un paquete en R se está modificando la ruta de búsqueda de *nombres* (de objetos y de funciones).

Ruta actual:

```
search()
```

```
## [1] ".GlobalEnv"        "package:knitr"     "package:stats"
## [4] "package:graphics"  "package:grDevices" "package:utils"
## [7] "package:datasets"  "Autoloads"         "package:base"
```

Cargamos un paquete:

```
library("MASS")
```

Volvemos a consultar la "ruta":

```
search()
```

```
##  [1] ".GlobalEnv"        "package:MASS"      "package:knitr"
##  [4] "package:stats"     "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"  "Autoloads"
## [10] "package:base"
```

Documentación y recursos sobre los paquetes

- Listado de Paquetes activos (hay más de 12.000!)aquí
- Listado de paquetes por áreas aquí
- Paquetes más descargados aquí
- Microsoft R Application Network aquí
- Paquetes huérfanos de mantenedor aquí

Estructura de la Presentación

Helping Yourself

```
help(lm)                    # Comando básico de búsqueda
?lm                         # ídem anterior
help("+")                   # Ayuda sobre un operador
help.search("norm")         # Busca entre los paquetes instalados las
                            # funciones que contienen el termino "norm"
??norm                      # ídem anterior
apropos("mean")             # Listado de funciones que contienen el termino "norm"
help(rlm, package="MASS")   # Busca la ayudad de una función especifica
                            # de un paquete
RSiteSearch("glm")          # Busca en los manuales y ficheros de ayuda de la web
                            # R-project
find("mean")                # Devuelve el paquete al que pertenece la función.
                            # (el paquete debe estar cargado: library())
example(contour)            # Ejecuta los ejemplos disponibles de la función
browseVignettes()           # Muestra en web las viñetas disponibles y el acceso
                            # a ellas. De un paquete: (package="package-name")
vignette(all=TRUE)          # Muestra todas las viñetas disponibles en pantalla
```

Ejemplo:

```
help(mean)
```

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

`x`
: An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.

`trim`
: the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

`na.rm`
: a logical value indicating whether `NA` values should be stripped before the computation proceeds.

`...`
: further arguments passed to or from other methods.

Value

If `trim` is zero (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If `x` is not logical (coerced to numeric), numeric (including integer) or complex, `NA_real_` is returned, with a warning.

If `trim` is non-zero, a symmetrically trimmed mean is computed with a fraction of `trim` observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

`weighted.mean`, `mean.POSIXct`, `colMeans` for row and column means.

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

Elementos de ficha técnica de una función

`Name`:	El nombre de la función, entre llaves {} el paquete al que pertenece la función
`Description`:	Breve descripción de la la función
`Usage`:	Sintaxis de la función
`Arguments`:	Explicación de los argumentos de la función
`Value`:	Características de la salida de la función
	(puede ser un objeto como un valor)
`References`:	Bibliografía relacionada con la función
`See Also`:	Funciones relacionadas.
`Examples`:	Ejemplos de cómo se usa la función

Ayuda externa

La comunidad de R es muy activa, por lo que en la web es muy fácil encontrar en cualquier buscador la solución a los problemas que se generen en un flujo de trabajo.

De manera centralizada se destacan estos sitios:

- → http://search.r-project.org/
- → https://www.rdocumentation.org/
- → https://rseek.org/
- → https://stackoverflow.com/questions/tagged/r
- → https://stats.stackexchange.com/questions/tagged/r

Estructura de la Presentación

Comandos básicos (I)

```r
ls()                          # lista los objetos que hay en memoria en ".GlobalEnv"
```

```
## [1] "X"     "media"
```

```r
y <- c(1, 5, 9)               # creo un objeto que contiene 3 números
ls()
```

```
## [1] "X"     "media" "y"
```

```r
rm(y)                         # elimina (remove) el objeto 'y'
objects()                     # ídem ls()
```

```
## [1] "X"     "media"
```

```r
rm(list=ls(all=TRUE))         # elimina todos los objetos de la sesión de trabajo
ls()
```

```
## character(0)
```

Algunos comandos básicos (II)

```
setwd()                          # establece directorio de trabajo
getwd()                          # devuelve la ruta actual
dir()                            # lista los archivos en el directorio
list.files()                     # ídem anterior
dir.create()                     # crea una carpeta en el directorio
```

Ejemplo:

```
# OPCION 1
setwd("C:/Users/usuario/Desktop/Curso_R")              # Usar: '\\' o '/'
# OPCION 2
directorio1 <- "C:/Users/usuario/Desktop/Curso_R"
setwd(directorio1)
dir.create("../Curso_R/Plots")
directorio2 <- "../Curso_R/Plots"
setwd(directorio2)
```

Este flujo de trabajo tiene sus problemas (ver esta discusión y estos paquetes : 'here' y 'rprojroot')

options()

Esta función permite modificar varias opciones sobre la forma en la que R calcula y muestra resultados.

```
names(options())
```

```
##  [1] "CBoundsCheck"             "HTTPUserAgent"            "OutDec"                   "PCRE_limit_recursion"
##  [5] "PCRE_study"               "PCRE_use_JIT"             "add.smooth"               "bitmapType"
##  [9] "browser"                  "browserNLdisabled"        "check.bounds"             "citation.bibtex.max"
## [13] "continue"                 "contrasts"                "defaultPackages"          "demo.ask"
## [17] "deparse.cutoff"           "device"                   "device.ask.default"       "digits"
## [21] "dvipscmd"                 "echo"                     "editor"                   "encoding"
## [25] "example.ask"              "expressions"              "help.search.types"        "help.try.all.packages"
## [29] "internet.info"            "keep.source"              "keep.source.pkgs"         "knitr.in.progress"
## [33] "locatorBell"              "mailer"                   "matprod"                  "max.print"
## [37] "menu.graphics"            "na.action"                "nwarnings"                "pager"
## [41] "papersize"                "pdfviewer"                "pkgType"                  "printcmd"
## [45] "prompt"                   "repos"                    "rl_word_breaks"           "scipen"
## [49] "show.coef.Pvalues"        "show.error.messages"      "show.signif.stars"        "showErrorCalls"
## [53] "str"                      "str.dendrogram.last"      "stringsAsFactors"         "texi2dvi"
## [57] "tikzDocumentDeclaration" "tikzMetricsDictionary"    "timeout"                  "ts.S.compat"
## [61] "ts.eps"                   "unzip"                    "useFancyQuotes"           "verbose"
## [65] "warn"                     "warning.length"           "width"
```

options()

Ejemplo:

```
100/654                          # ver valor por defecto: options("digits")

## [1] 0.1529052

options(digits = 4)
100/654                          # ver valor actual: getOption("digits")

## [1] 0.1529

set.seed(1234)
rnorm(8)

## [1] -1.2071  0.2774  1.0844 -2.3457  0.4291  0.5061 -0.5747 -0.5466

options(width = 40)
set.seed(1234)
rnorm(8)

## [1] -1.2071  0.2774  1.0844 -2.3457
## [5]  0.4291  0.5061 -0.5747 -0.5466
```

Errores y advertencias en R

Importante: que no se imprima en pantalla un mensaje de `Error` o un `Warning message` no significa que lo que estamos haciendo está bien!.

La principal diferencia entre un error y una advertencia es que el error no produce ningún resultado, mientras que una advertencia si.

Ejemplo:

```
mean(variable.1)

## Error in mean(variable.1): object 'variable.1' not found

variable.1 <- seq(1, 100, 0.3)
mean(variable1)

## Error in mean(variable1): object 'variable1' not found

mean(variable.1)

## [1] 50.5
```

$\Rightarrow$ El primer error indica que el objeto `variable.1` no existe.

$\Rightarrow$ El segundo error indica lo mismo, pero el problema es que el objeto llamado `variable.1` está mal escrito.

Ejemplo:

```
sqt(8)

## Error in sqt(8): could not find function "sqt"

sqrt(8)

## [1] 2.828
```

$\Rightarrow$ Este error indica que la función `sqt` no existe. El problema es que la función está mal escrita.

Uso de funciones: introducción necesaria

Una función en R es una o varias sentencias de código que realizan una operación determinada.

Una función tiene dos características fundamentales para su correcto uso: un nombre y argumentos que siembre van entre paréntesis.

Ejemplo: función para obtener el promedio de un conjunto de datos

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Nombre de la función: `mean`
Argumentos de la función: `x, trim, na.rm, ...`

$\Rightarrow$ De los 4 argumentos hay dos que tienen un valor asignado mediante el signo '='. Estos son valores por defecto. Si el usuario no los modifica la función siempre va a asumir esos valores.

Uso de funciones: introducción necesaria

Cuando se usa una función:

- Los argumentos que tienen valores por defecto pueden no llamarse cuando se ejecuta la función.
- Los argumentos tienen un orden que resulta importante únicamente si se opta por no nombrarlos.
- Los argumentos pueden ir en distinto orden siempre que sean nombrados y asignados.

Ejemplo: todas estas opciones generan el mismo resultado

```
mean(x = 1:10)
mean(1:10)
mean(x = c(1:10, NA), trim = 0, na.rm = TRUE)
mean(trim = 0, x = c(1:10, NA), na.rm = TRUE)
mean(c(1:10, NA), 0, TRUE)
```