

# Curso de R y estadística básica

[Felipe de J. Muñoz González]

[fmunoz@lcg.unam.mx](mailto:fmunoz@lcg.unam.mx)

Introducción  
Descargar Presentación

# Temario de esta presentación

- Introducción a R
  - Historia de R
  - Instalación de R
  - R, R commander y RStudio
  - Bibliografía
- Fundamentos de R y programación
  - Expresiones y sintaxis en R
  - Funciones en R
  - Crear una función en R
  - Ayuda, manuales y ejemplos
  - Manejo de archivos del sistema
- Variables y estructuras de datos
  - El vector como unidad básica de R
  - Función concatenar
  - Generadores de secuencias
  - Acceso a elementos de un vector
  - Modificación de elementos de un vector
  - Nombres del vector
  - Operaciones sobre vectores
  - Listas y sus operaciones

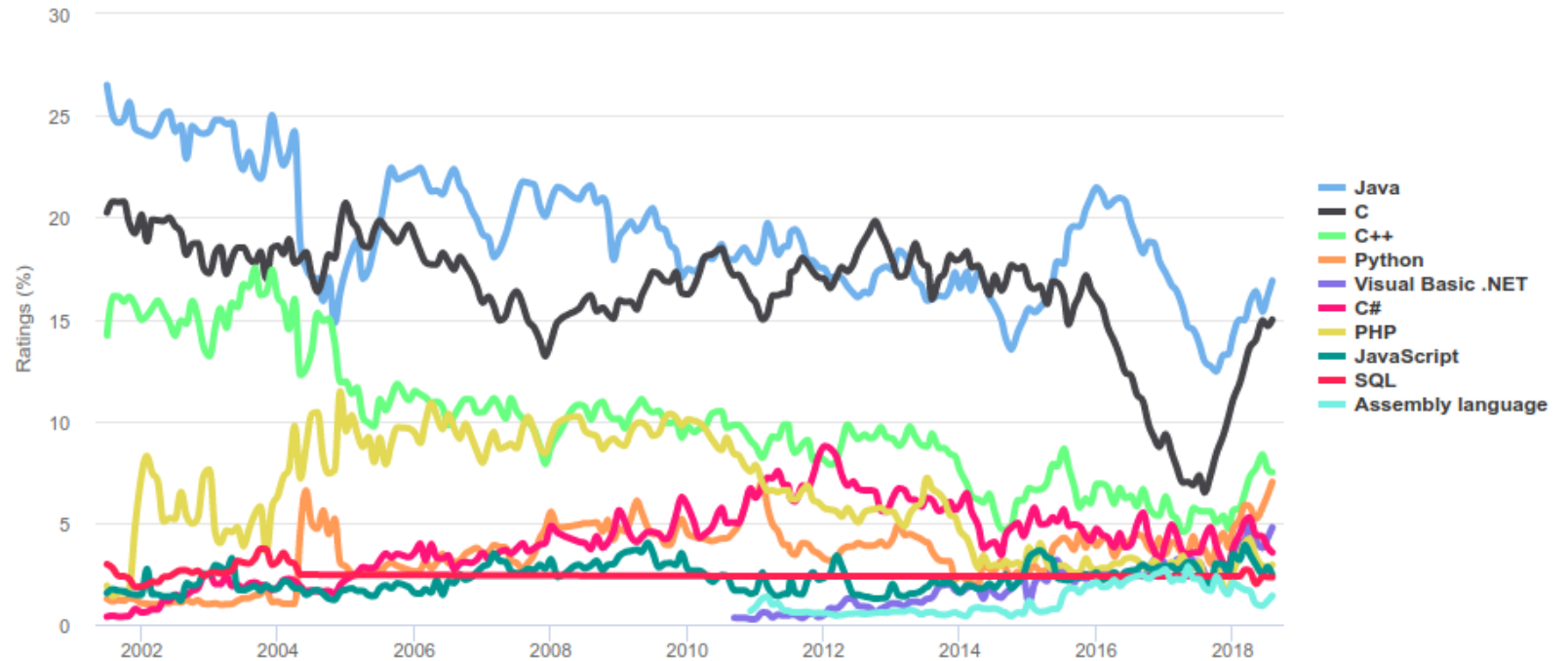
# Breve historia

- Creación de S por parte de AT&T en Nueva Jersey (John Chambers & Rick Becker)
- Se mezcla S y Scheme para generar R en la universidad de Auckland (Robert Gentleman & Ross Ihaka, 1995)
  - R esta escrito en C y Fortran
  - GNU General Public License (freely available)
- Creación del R Development Core Team y el CRAN (1997)
- Versión mas antigua. Versión 0.49 (23 de abril de 1997)
- Versión 3.4.0 (21 de abril de 2017)

```
> contributors()
```

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# ¿Qué es R?



**R es un lenguaje de programación para la estadística y el modelado de datos.**

**R tiene las siguientes características:**

- Es elegante y versatil
- Sintaxis diseñada para trabajar con datos
- Capacidades gráficas altamente potentes
- Facil y eficiente manipulación de datos
- Contiene herramientas enfocadas a las areas biológicas.

*\* R es lo que requieres para ciencias biológicas*

## Why R is Not Enough

- The R interpreter is not fast. Execution of large amounts of R code can be unacceptably slow.
- R is set up to carry out vectorised computations and not scalar (element-by-element) computations.
- R was designed to hold its data “in core” and this places severe limitations of the size of problem which can be dealt with.

## Directions for Future Work

- Wait for faster machines.
- Introduce more vectorisation and take advantage of multicores.
- Make changes to R to eliminate bottlenecks.
- Sweep the page clean and look at designs for new languages.

\* R: Lessons Learned, Directions for the Future *Ross Ihaka*



# Instalación

## - Windows

Lo primero es identificar donde descargamos R para instalarlo:

<https://cran.r-project.org/bin/windows/base/>

---

### R-3.3.2 for Windows (32/64 bit)

[Download R 3.3.2 for Windows](#) (62 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the [md5sum](#) of the .exe to the [true fingerprint](#). You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

#### Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

#### Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [<CRAN MIRROR>/bin/windows/base/release.htm](#).

---

Last change: 2016-10-31, by Duncan Murdoch



# Instalación

## - MAC

Lo primero es identificar donde descargamos R para instalarlo:

<https://cran.r-project.org/bin/macosx/>

# Instalación

## - Linux

Abrimos una terminal de linux (Ctrl + Alt + T) y dentro de esta, dependiendo del sistema operativo:



Ubuntu

```
$ sudo apt-get install r-base
```



Fedora

```
$ su -c 'yum install R'
```



Arch Linux

```
$ sudo pacman -S r
```

# ¿Cómo funciona?

- Ejecutar desde el cmd/terminal

- Entornos graficos

## Desde Windows:

- Opcion A:

Inicio > Simbolo del sistema

- Opcion B:

Buscar > CMD

## Desde MAC/Linux:

1. Se abre la terminal

## Se ejecuta:

```
$ R
```

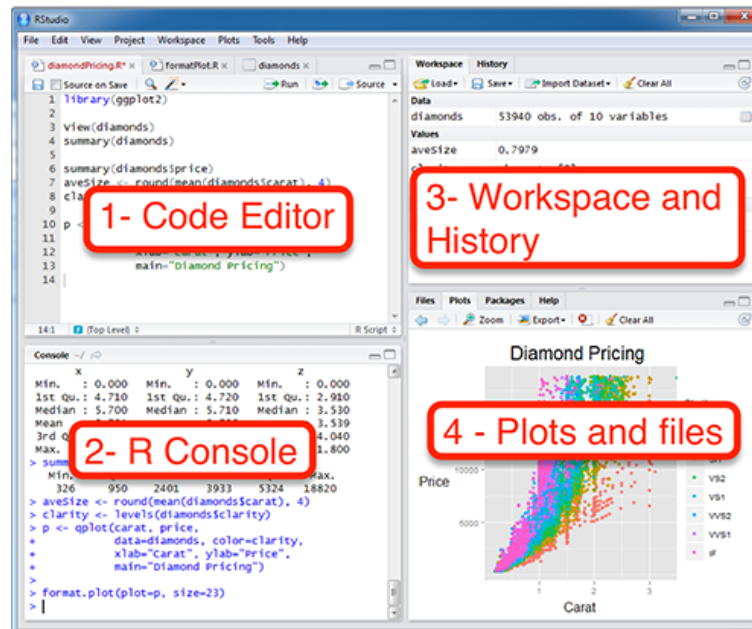
# ¿Cómo funciona?

## - Entornos graficos

## Rstudio

Ambiente gráfico integrado, se basa en diversos compartimentos:

- Consola para editar código
- Ventana de datos e historial
- Ventana de la Consola
- Ventana de gráficas y archivos





## Importando desde el ambiente grafico

The screenshot shows the R Studio Environment pane. The 'Import Dataset' menu is open, displaying options: 'From CSV...', 'From Excel...', 'From SPSS...', 'From SAS...', and 'From Stata...'. The 'Data' section lists several datasets with their respective variable counts: 'data' (2 variables), 'expri' (14 variables), 'fy201' (18 variables), 'wages' (12 variables), and 'x1000' (18 variables). The 'Values' section shows the following settings:

| Variable | Value  |
|----------|--|
| destfile | "fy2015f_4050_final.xls"                         |
| i        | 30000L   |
| m        | 1L   |
| url      | "http://www.huduser.org/portal/datasets/fmr/..." |
| x        | "car"  |

# ¿Cómo funciona?

## - Entornos graficos

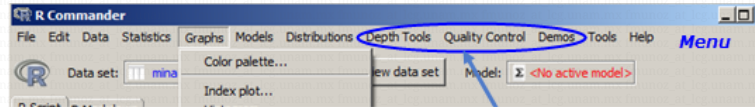
### R commander (Rcmdr)

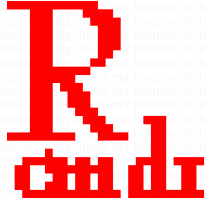
Es una interfaz gráfica que cuenta con botones y menus extensos, las características son

- Contiene codigos precargados (SPSS, SAS o Stata)
- No provee acceso directo a la linea de comandos de R
- No es enriquecido gráficamente, contiene 3 paneles:
  - Ventana del script (código ejecutandose)
  - Ventana de Salida (Imprime los resultados)
  - Ventana de Mensajes (Errores/advertencias/notas)

# ¿Cómo funciona?

## - Entornos graficos





## Utilizando las herramientas predefinidas

The screenshot shows the Rcmdr application window. The 'Statistics' menu is open, displaying options like Summaries, Contingency tables, Means, Proportions, Variances, Nonparametric tests, Dimensional analysis, and Fit models. The 'Fit models' option is selected, showing a submenu with Linear regression..., Linear model..., Generalized linear model..., Multinomial logit model..., and Proportional-odds logit model.... A 'Submit' button is visible next to the 'Proportional-odds logit model...' option.

The 'Script Window' contains the following R code:

```
x<-rnorm(100)
y<-rnorm(100)
data1<-data.frame(x,y)
attach(data1)
Dataset <- edit(as.data.frame(NULL))
```

The 'Output Window' shows the execution of the code:

```
> y<-rnorm(100)
> x<-rnorm(100)
> data1<-data.frame(x,y)
> attach(data1)
> Dataset <- edit(as.data.frame(NULL))
```

The 'Messages' window at the bottom displays an error message: 'ERROR: empty data set.' and a note: 'NOTE: The dataset data1 has 100 rows and 2 columns.'

The 'One-Way Analysis of Variance' dialog box is open, showing the 'Enter name for model: AnovaModel.1'. The 'Groups (pick one)' list includes CLASS\_DESCR, DEPT\_DESCR, GENDER, and LOCATION\_DESCR. The 'Response Variable (pick one)' list includes CostSum, DEPTID, FUND\_CODE, and TOTAL\_COST. The 'Pairwise comparisons of means' checkbox is checked. The 'OK' button is highlighted.

Annotations:

- Variable selection window, allows choosing from list**: Points to the 'Groups (pick one)' list in the ANOVA dialog.
- Sensible defaults with the option of customization for expert users**: Points to the 'Response Variable (pick one)' list in the ANOVA dialog.
- Checkboxes for access to common options for a statistical test**: Points to the 'Pairwise comparisons of means' checkbox in the ANOVA dialog.



## ¿Cuál usar y como instalarlo?

# Comparación

## RStudio

- Provee acceso directo al código en R.
- Uso para proyectos que requieren interacción directa con el código o manipulación de datos compleja

## Rcmdr

- Simple y amable para el usuario sobre todo en análisis estadísticos y diagnósticos.
- Uso para análisis tradicionales, datos convencionales y tests estadísticos.

**NOTA:** Es posible ejecutar Rcmdr desde R-Studio.

# Instalación de R-Studio

RStudio tiene diferentes versiones:

- Version gratis para escritorio
- Version de paga para escritorio
- Version gratis para servidor
- Version pro para servidor

Para descargarlo entramos a

<https://www.rstudio.com/products/rstudio/download/>

**RStudio Desktop 1.0.136** — [Release Notes](#)

RStudio requires R 2.11.1+. If you don't already have R, download it [here](#).

## Installers for Supported Platforms

| Installers   | Size    | Date       | MD5                              |
|--|---------|------------|----------------------------------|
| <a href="#">RStudio 1.0.136 - Windows Vista/7/8/10</a>                         | 81.9 MB | 2016-12-21 | 93b3f307f567c33f7a4db4c1140993be |
| <a href="#">RStudio 1.0.136 - Mac OS X 10.6+ (64-bit)</a>                      | 71.2 MB | 2016-12-21 | 12d6d6ade0203a2fcef6fe3dea65c1ae |
| <a href="#">RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (32-bit)</a>             | 85.5 MB | 2016-12-21 | 0a20fb89d8aaeb39b329a640ddadd2c5 |
| <a href="#">RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (64-bit)</a>             | 92.1 MB | 2016-12-21 | 2a73b88a12a9fbaf96251cecf8b41340 |
| <a href="#">RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)</a> | 84.7 MB | 2016-12-21 | fa6179a7855bf0f939a34c169da45fd  |
| <a href="#">RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)</a> | 85.7 MB | 2016-12-21 | 2b3a148ded380b704e58496befb55545 |

## Zip/Tarballs

| Zip/tar archives   | Size     | Date       | MD5                              |
|--|----------|------------|----------------------------------|
| <a href="#">RStudio 1.0.136 - Windows Vista/7/8/10</a>                         | 117.5 MB | 2016-12-21 | f415939bf5012c0ab127c7cfbc9600be |
| <a href="#">RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (32-bit)</a>             | 86.2 MB  | 2016-12-21 | fca75f953dd425694b7fd4335bd29165 |
| <a href="#">RStudio 1.0.136 - Ubuntu 12.04+/Debian 8+ (64-bit)</a>             | 93.2 MB  | 2016-12-21 | 7cf0092653aa44fc76325a8f1325fb1f |
| <a href="#">RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)</a> | 85.4 MB  | 2016-12-21 | 30c89299d30ec03b38098e51e9bf49b8 |
| <a href="#">RStudio 1.0.136 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)</a> | 86.6 MB  | 2016-12-21 | ea2a262f650e92f568f48edc1c093902 |

## Source Code

A tarball containing source code for RStudio v1.0.136 can be downloaded from [here](#)

# Instalación de Rcmdr

Ejecutamos R

```
[usuario@equipo ~]$ R
```

Instalamos el paquete de Rcmdr

```
> install.packages("Rcmdr",dependencies=TRUE)
```

Seguimos las instrucciones de la salida

```
Aviso en install.packages("Rcmdr", dependencies = TRUE) :  
'lib = "/usr/lib/R/library"' is not writable  
Would you like to use a personal library instead? (y/n) y  
Would you like to create a personal library  
~/R/x86_64-unknown-linux-gnu-library/2.15  
to install packages into? (y/n) y
```

Se abrirá una ventana para seleccionar el repositorio de dónde descargar los paquetes necesarios. Seleccionamos el que queramos y después de aceptar empezará a descargar los paquetes.



```
chooseCRANmirror(graphics=FALSE) # chooseCRANmirror(81)
```

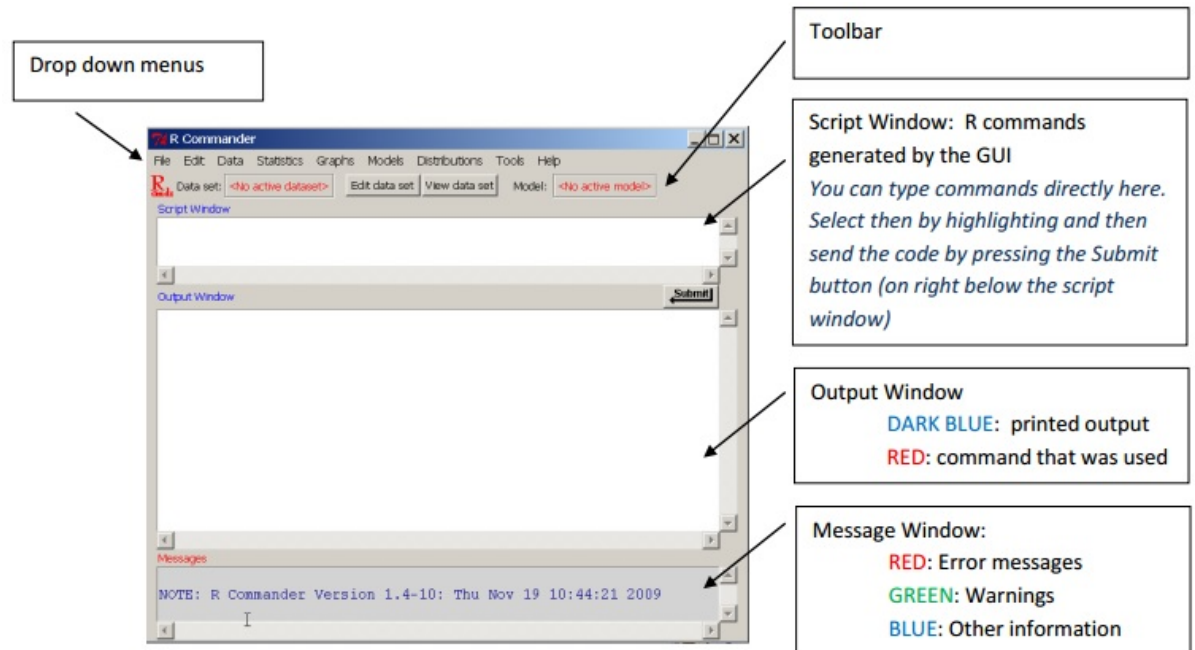
# Ejecutar Rcmdr

## Ejecutamos R

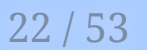
```
[usuario@equipo ~]$ R
```

## Cargamos la libreria de Rcmdr y ejecutamos

```
> library("Rcmdr")
```



# Mis primeros pasos en R



# Sintaxis

```
> 1+2 # Sumar 1 más 2
```

## - Aritmetica

```
> 6*5+3 # 6 por nueve añadir 3
```

```
> 2**3 #2 elevado al exponente 3  
> 2^8 #2 elevado al exponente 8
```

```
> log(exp(3)) # e elevado a la 3 y logaritmo del resultado  
> log10(1000) #Logaritmo de 1000 base 10  
> log2(10**20) # Logaritmo de 10 elevado a la 20 base 10
```

# Sintaxis

## - Aritmetica

```
> 10/3  
> options(digits =16) #ver mas digitos  
> 10/3
```

```
> pi #numeros irracionales
```

```
> exp(1)
```



# Sintaxis

## - Aritmetica

## - Texto

```
> "Esto es una cadena de texto!"
```

```
> palabras<-c("Esto", "es", "un", "conjunto", "de", "palabras")
```

# Sintaxis en R

## - Aritmetica

```
> 3 > 4 # ¿Es 3 mayor que 4?
```

## - Texto

## - Operaciones Logicas

```
> T == TRUE #¿Es T igual a TRUE?
```

# Sintaxis en R

## - Aritmetica

## - Texto

## - Operaciones Logicas

## - No definidos

```
> sqrt(-1)
```

```
> sqrt(-1+0i)
```

```
> sqrt(as.complex(-1))
```

**NaN** Not a Number

# Variables

# Variables

## - Definición

### Definición de una variable

**Estructura de datos** que pueden cambiar de contenido a lo largo de la ejecución de un programa.

Una variable requiere

- Un espacio en el sistema de almacenaje
- Un identificador (nombre) asociado a ese espacio
- Cantidad o informacion ya sea conocida o desconocida (Valor)

# Variables

## - Definición

## - Tipos

### Tipos de Variables:

- **Tipo de dato cuantitativo.**
  - Numerico entero (Discreto)
  - 0, 1, -2, 3, 10, ...
  - **Numerico flotante/doble (Continuo)**
  - 12.1, 200, 5.2E10, 0.1232, pi, e, sqrt(2)
- **Tipo de dato cualitativo(categórico).**
  - Tipo de dato de texto

```
> str(state.abb)  
> "Se pueden usar simbolos.,-.$%&/()="
```

# Variables

## - Definición

## - Tipos

### Tipos de Variables:

- **Tipo de dato lógico.**

> TRUE/FALSE, T/F

- **Datos faltantes.**

> NA  
> NaN

# Variables

## - Definición

Crea una variable que tenga el valor 42 que se llame a y dividela entre 3

```
> 1 <- 42 #a = 42  
> var_1/3
```

## - Tipos

Crea una cadena de texto y ponla en la variable txt

## - Creación

```
> txt <- "Arrg, Soy un pirata"
```

Asigna un valor de VERDADERO a una variable logic

```
> z <- TRUE
```



# Variables

## - Definición

## - Tipos

## - Creación

## Datos Faltantes

```
> x <- c(3, 7, NA, 4, 7)
> y <- c(5, NA, 1, 2, 2)
> x + y
> sum(x)
> sum(x, na.rm = TRUE)
> is.na(x)
```

# Variables

## Imprime los valores de txt, logic y de a

### - Definición

```
> txt
```

### - Tipos

### - Creación

```
> logic
```

```
> a
```

# Variables

## - Definición

## - Tipos

## - Creación

Los nombres de las variables solo pueden tener (Nombres significativos):

- [a-zA-Z]+
- [0-9]+
- [.\_]+

Que cosas **NO** pueden tener:

- Numeros para iniciar
- Simbolos seguidos de numeros (Ej. A+3)
- Solamente numeros
- Solamente simbolos

# Vectores

## - Introducción a vectores

### Unidad básica de variable en R.

### Introducir 74,31,95,61,76,34,23,54,96 en R

#### 1. Funcion c

```
x <- c(74,31,95,61,76,34,23,54,96) #Numerico
```

```
x <- c("74",31,95,61,76,34,23,54,96) #String
```

# Vectores

## - Introducción a vectores

### 1. Funcion scan

```
x <- scan() # empty to end
```

# Vectores

## - Introducción a vectores

### 1. Repeated data; Regular patterns

```
> seq(from = 1, to = 5)
```

```
> x <- seq(from = 2, by = -0.1, length.out = 20)  
> y <- seq(from = 2, to= 100, by = 5)
```

```
> 1:5
```

# Vectores

```
> x[1]
```

## - Introducción a vectores

```
> x[c(1, 3, 4, 8)]
```

## - Indexación

```
> x[-c(1, 3, 4, 8)]
```

```
> LETTERS[1:5]
```

```
letters[-(6:24)]
```

# Listas

## - Introducción y construcción de Listas

- Una lista es una colección de elementos que pueden ser de distintos tipos y que generalmente están identificados por un nombre.
- Para crear una lista se utiliza la función list

```
> lst <- list(hombre = "Pedro", mujer = "María", casados = T)
```



# Listas

## - Introducción y construcción de Listas

## - Accesores y modificadores

```
> lst$hombre
```

```
> lst[c("hombre", "número.hijos")]
```

```
> lst[c(TRUE, FALSE, FALSE, TRUE, FALSE)]
```

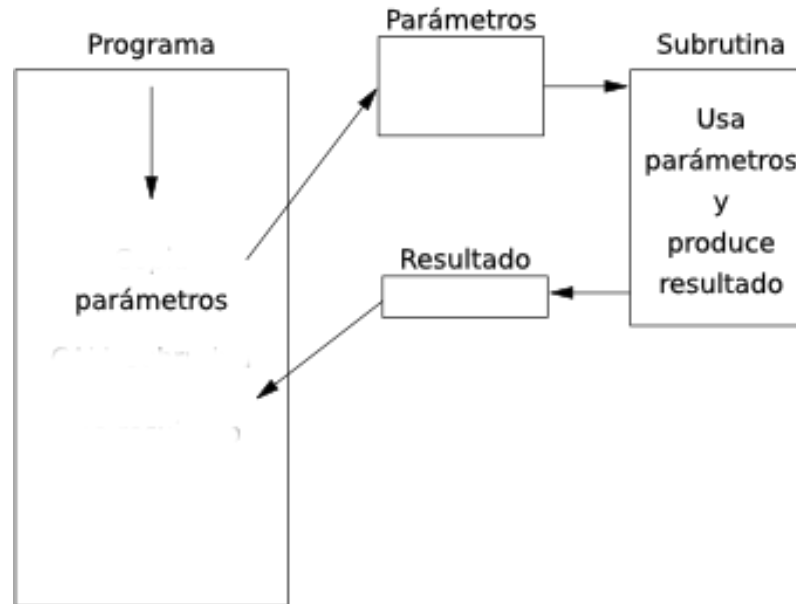
```
> lst[c(1, -4)]
```

# Funciones

## - Definición

Definición:

**Subrutina** o **subprograma** (también llamada procedimiento, función o rutina), que se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica.



# Funciones

## - Definición

## - Ejemplos

```
> sum(1,2,6,7,9)
```

Algunas funciones requieren argumentos los cuales tienen nombres determinados

```
> rep("Yo ho! un pirata soy", times=3)
```

**Ejercicio:** Obten la raíz de 16 utilizando la función sqrt

# Funciones

## - Definición

```
> mi_función<-function(varA,varB,varC,...){  
>   #AQUÍ VA EL CÓDIGO A EJECUTAR  
>   # ...  
>   #AQUÍ HAY MÁS CÓDIGO  
> }
```

## - Ejemplos

Algunas funciones requieren argumentos los cuales tienen nombres determinados

## - Creación

```
> rep("Yo ho! un pirata soy", times=3)
```

# Funciones

## - Definición

**Ejercicio:** Genera una funcion de fibonacci.

$$1. F\{n\}=F\{n-1\}+F\{n-2\}$$

$$2. F\{1\}=1, \text{ ; } F\{2\}=1 \text{ } \text{ or } F\{0\}=0, \text{ ; } F\{1\}=1$$

## - Ejemplos

## - Creación

# Funciones

## - Definición

## - Ejemplos

## - Help, example

De todas las funciones, la que debemos de considerar como básica es **help**

```
> help (sum)
```

```
> ?sum
```

Otra es la función **example**

```
> example(min)
```

**Ejercicio:** Obten el help de la función rep

# Funciones

## - Definición

## - Ejemplos

## - Help, example

## - Código

Escribe el nombre de una función sin parentesis

```
> intersect
```

Buscar una función con methods

```
> methods(summary)
```

Ejercició. Busca la ayuda y el metodo de un test de wilcox

```
> wilcox.test  
> ?wilcox.test  
> methods(wilcox.test)
```

Ejercició. busca las funciones de R que comienzan con  
lengt

```
> ??lengt
```

# Obtener ayuda

## - Interna

- **FAQ on R:** <https://cran.r-project.org/doc/FAQ/R-FAQ.html>
  - **FAQ on R for Windows:** <https://cran.r-project.org/bin/windows/base/rw-FAQ.html>
  - **R Manuals:** <https://cran.r-project.org/manuals.html>
  - **R functions (text):** `help("mean")`, `?plot`
  - **HTML Help:** `help.start()`
  - **Search help:** `help.search("plo")` // ?? plo
  - **search.r-project.org**
  - **R Help Mailing Lists** <http://www.r-project.org/mail.html>



# Obtener ayuda

- Interna

- Externa

## Para R

- **R-Forge:** <http://r-forge.r-project.org/>
- **R Wiki:** <http://wiki.r-project.org/rwiki/doku.php>
- **R Graph Gallery:** <http://addictedtor.free.fr/graphiques/>
- **R Graphical Manual**  
<http://bm2.genes.nig.ac.jp/RGM2/index.php>
- **RSeek** (<http://www.rseek.org>)

## Para estadística

- **Probabilidad y Estadística para ingeniería y ciencias**  
*Warpole • Myers • Myers. 9th Ed*

## Para todo

- **Google** [www.google.com](http://www.google.com)

# Obtener ayuda

- Interna

- Externa

- Otros tips

- **Console: Shortcuts (Ej. Ctrl+L)**
  - **Up Arrow** Para comandos utilizados
  - **objects()** o **ls()** Variables actuales
  - **remove(var1,var2,var3)** o **rm(var1,var2)** Borrar variables
  - **rm(list=ls())** Borrar todo

# Archivos

- Ver archivos,  
obtener el  
directorio y  
cambiar de  
directorio

Con el fin de leer una carpeta dentro del sistema podemos utilizar la función **list.files()**

```
> list.files()
```

Localizar el directorio actual

```
> getwd() ##Get* *W*orking *D*irectory
```

Cambiar el directorio actual

```
> setwd() ##Set* *W*orking *D*irectory
```

# Archivos

- Ver archivos

- Scripts R

Cuando se programa en R, se generan documentos llamados **Scripts**, los cuales en general por convención finalizan en .R y pueden ser ejecutados en el momento que se desee.

Si se quiere ejecutar un código determinado desde la terminal se utiliza la función **source**:

[beers\\_code.r](#)

```
> source("beers_code.r")
```

That's all folks (for now)!