



Análisis de datos con R

Pablo Fernández Navarro



PARTE A:

- Introducción a R
- Packages
- Importación
- Manejo básico de datos
 - Data frame
 - Fechas y Caracteres
 - Combinación y reestructuración de data.frames/data.tables
- Análisis bioestadístico básico
- Exportación / Generación de informes

PARTE B:

- Aplicación práctica de análisis cualitativo/cuantitativo

PARTE A

Introducción a R

Parte A: Introducción a R

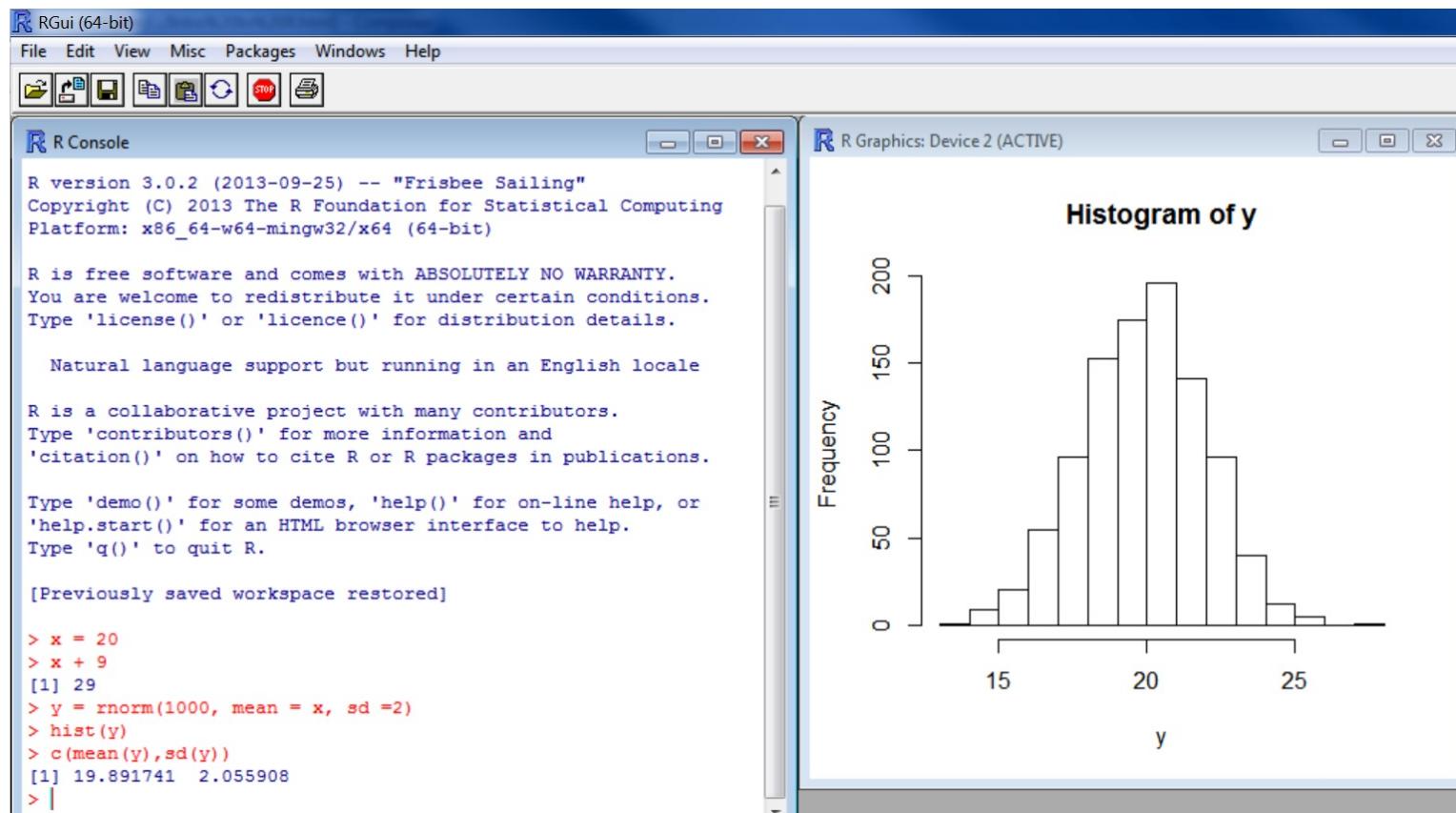
- Que es R
- Carácterísticas
- ¿Por qué R?
- Enlaces
- Libros y manuales
- Comunidad R / Actualidad
- Obtención e instalación
- Inicio de una sesión en R
- Ayuda
- Editores
- Interfaces (RStudio)

Introducción a R: ¿Qué es R?

- R es un lenguaje y entorno de programación para análisis estadístico y gráfico gratuito (“high-level language”).
- Es un lenguaje de programación orientado a objetos.
- Proyecto de software libre, resultado de la implementación GNU del premiado lenguaje “S” y “Scheme”.
- R y S-Plus-versión comercial de S son, probablemente, los dos lenguajes más utilizados en investigación por la comunidad estadística.
- Muy populares en el campo de la investigación biomédica, la bioinformática y las matemáticas financieras.

Introducción a R: ¿Qué es R?

- The R environment: R is an integrated suite of software facilities for data manipulation, calculation and graphical display.



Introducción a R: Características

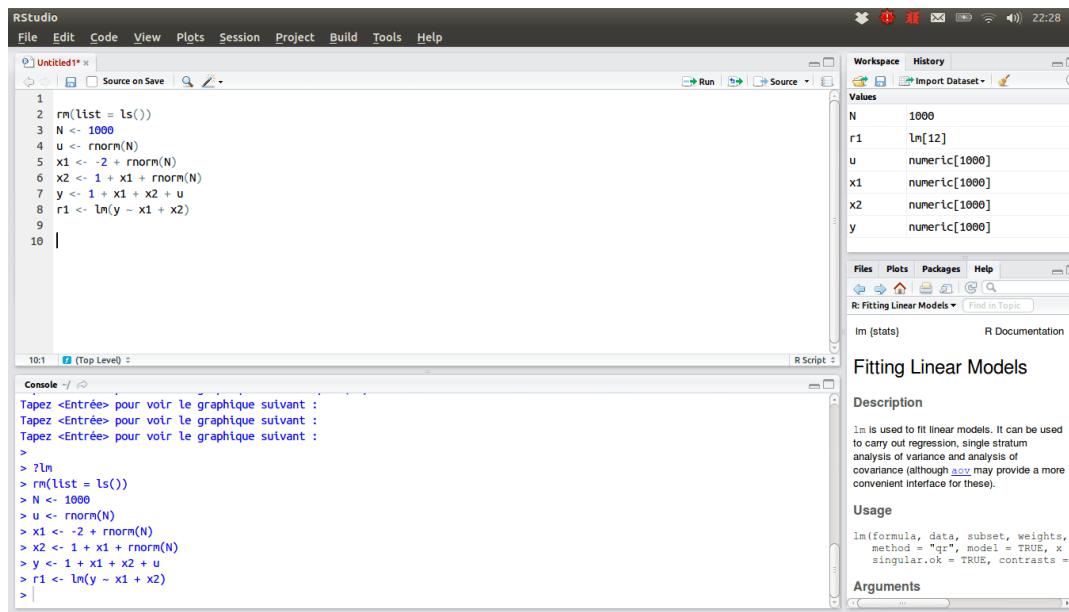
- Versatilidad: posibilidad de cargar diferentes paquetes con finalidades específicas de cálculo o gráfico, o utilización desde lenguajes de programación interpretados.

<https://cran.r-project.org/web/packages/index.html>

- R se distribuye bajo la licencia GNU GPL y está disponible para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux.
- Lenguaje interpretado y que está orientado a objetos.

Introducción a R: Características

- Puede integrarse con distintas bases de datos
- Existen diversas interfaces gráficas para R, como R-Commander, RStudio, etc.
- Se parece a Matlab y a Octave, y su sintaxis recuerda a C/C++.



Introducción a R: Open development software projects

- PBDR: Programming with Big Data in R. <http://www.r-pbd.org/>
- Bioconductor: Bioinformatics and Computational Biology using R. <http://www.bioconductor.org>
- RDM: Data Mining with R. <http://www.rdatamining.com/>
- Rmetrics: Computational finance) <https://www.rmetrics.org/>

Introducción a R: ¿Por qué R?

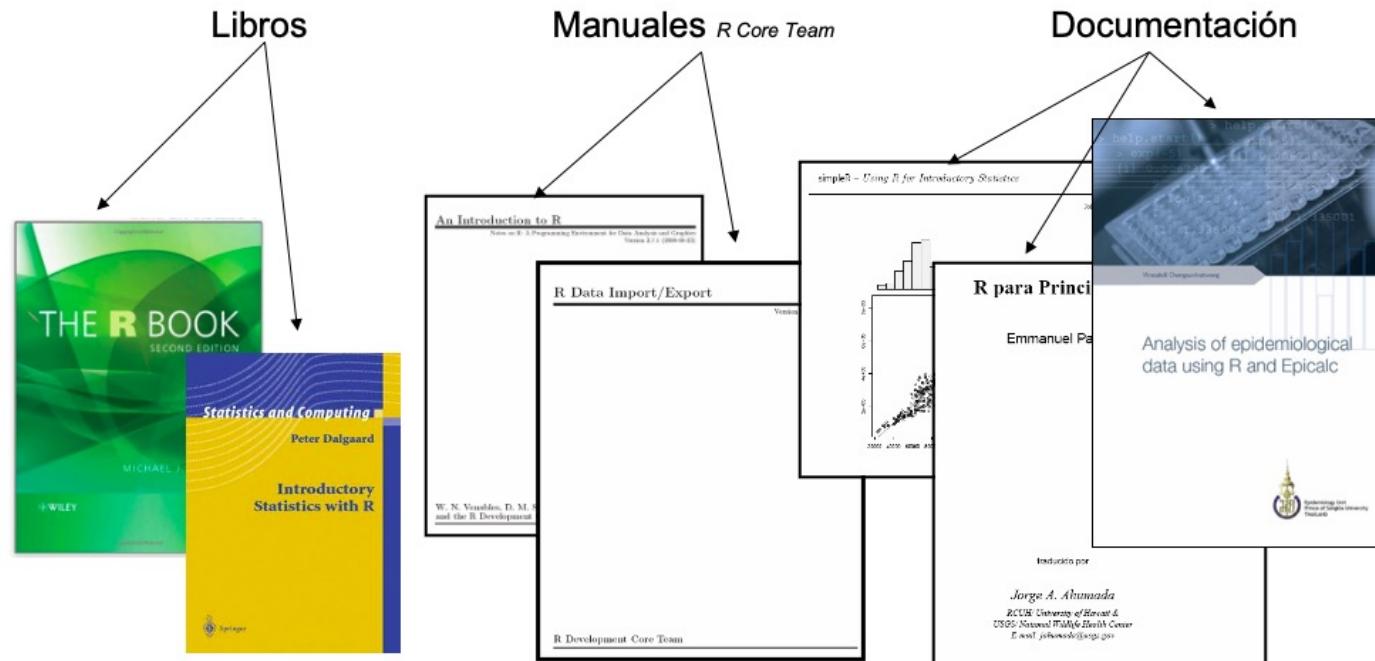
- Cobertura de análisis y disponibilidad de nuevas aplicaciones.
- Ser capaz de entender la literatura.
- Gran soporte.
- Capacidad de programar y construir funciones propias.
- No es de pago

Introducción a R: Enlaces

- El proyecto R:
<http://www.r-project.org/>
- R-Wiki:
<http://wiki.r-project.org/rwiki/doku.php>
- Interfaz Web para R:
<http://www.math.montana.edu/Rweb/>
- R Graph Gallery:
<https://r-graph-gallery.com/>

Introducción a R: Libros y manuales

- BOOKS: <https://www.r-project.org/doc/bib/R-books.html>
 - MANUALS: <https://cran.r-project.org/manuals.html>
 - DOCUMENTATION: <https://cran.r-project.org/other-docs.html>



Introducción a R: Comunidad R / Actualidad

useR! — International R User Conference



<https://www.r-project.org/conferences/>

Introducción a R: Comunidad R / Actualidad

- Burns Statistics Ltd., London, U.K.
- Department of Statistics, Brigham Young University, Utah, USA
- Paul von Eikeren, USA
- Institute of Mathematical Statistics (IMS), Ohio, USA
- Loyalty Matrix Inc., California, USA
- Mango Solutions, Chippingham, UK
- Marc Schwartz, USA
- Merck and Co. Inc., USA
- Numbers Internation Pty Ltd, Australia
- Prediction Company, Santa Fe, New Mexico, USA
- Saxo Bank, Denmark
- Schröder Investment Management Ltd., London, UK
- InterContinental Hotels Group, USA
- Shell Statistics and Chemometrics, Chester, UK
- Statisticon AB, Uppsala, Sweden
- Astra Zeneca R&D Mölndal, Mölndal, Sweden
- AT&T Labs, New Jersey, USA
- Baxter AG, Vienna, Austria
- Baxter Healthcare Corp., California, USA
- BC Cancer Agency, Vancouver, Canada
- Black Mesa Capital, Santa Fe, USA
- Boehringer Ingelheim Austria GmbH, Vienna, Austria
- Breast Center at Baylor College of Medicine, Houston, Texas, USA
- Center für digitale Systeme, Freie Universität Berlin, Germany
- Dana-Farber Cancer Institute, Boston, USA
- Department of Biostatistics, Johns Hopkins University, Maryland, USA
- Department of Biostatistics, Vanderbilt University School of Medicine, USA
- Department of Economics, Stockholm University, Sweden
- Department of Mathematics and Statistics, Utah State University, USA
- Department of Statistics, University of California at Los Angeles, USA
- Department of Statistics, University of Warwick, Coventry, UK
- Department of Statistics, University of Wisconsin-Madison, Wisconsin, USA
- Department of Statistics, Iowa State University, USA

- Department of Statistics & Actuarial Science, University of Iowa, USA
- Dipartimento di Statistica, Università Ca' Foscari di Venezia, Italy
- Division of Biostatistics, University of California, Berkeley, USA
- Ef-prime Inc., Tokyo, Japan
- European Bioinformatics Inst., UK
- Hygeia Associates, California, USA
- Lehrstuhl für Rechnerorientierte Statistik und Datenanalyse, University of Augsburg, Germany
- MPI for Demographic Research, Rostock, Germany
- Massachusetts General Hospital Biostatistics Center, Boston, USA
- National Public Health Institute, Helsinki, Finland
- Norwegian Institute of Marine Research, Bergen, Norway
- School of Economics and Finance, Victoria University of Wellington, New Zealand
- Spotfire, Massachusetts, USA
- TERRA Lab, University of Regina - Department of Geography, Canada
- ViaLactia Biosciences (NZ) Ltd, Auckland, New Zealand
- Adelchi Azzalini (Italy)
- AT&T Research (USA)
- Austrian Association for Statistical Computing (Austria)
- BC Cancer Agency, Vancouver (Canada)
- Fabian Barth (Germany)
- Biostatistics and Research Decision Sciences, Merck Research Laboratories (USA)
- Brian Caffo (USA)
- David W. Crawford (USA)
- Dianne Cook (USA)
- Yves De Ville (France)
- Department of Economics, University of Milano (Italy)
- Dipartimento di Scienze Statistiche e Matematiche di Palermo (Italy)
- Emanuele De Rinaldis (Italy)
- Zubin Dowlaty (USA)
- Faculty of Economics, University of Groningen (Netherlands)
- Jaimison Fargo (USA)

Introducción a R: Obtención e instalación

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

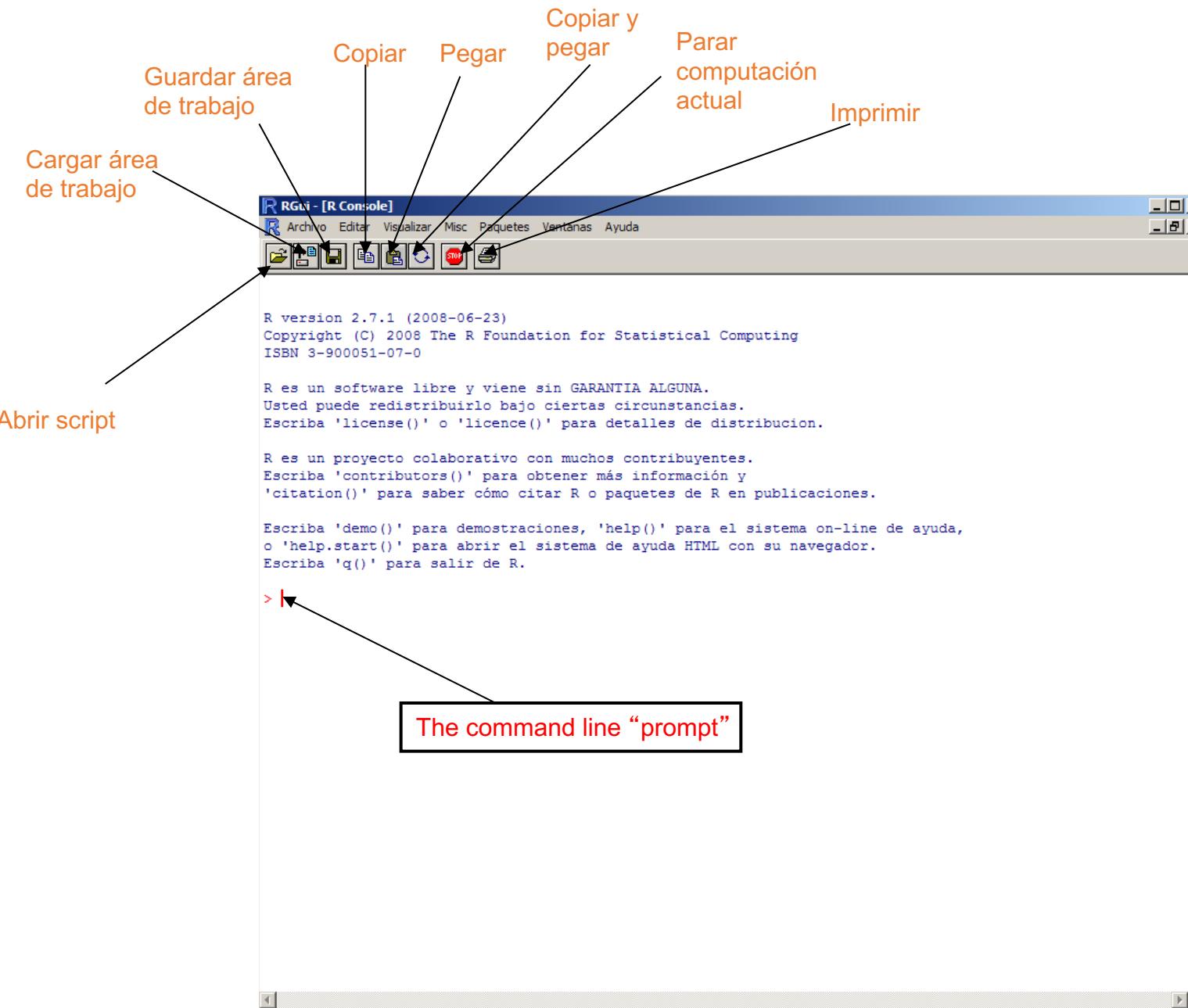
- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

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

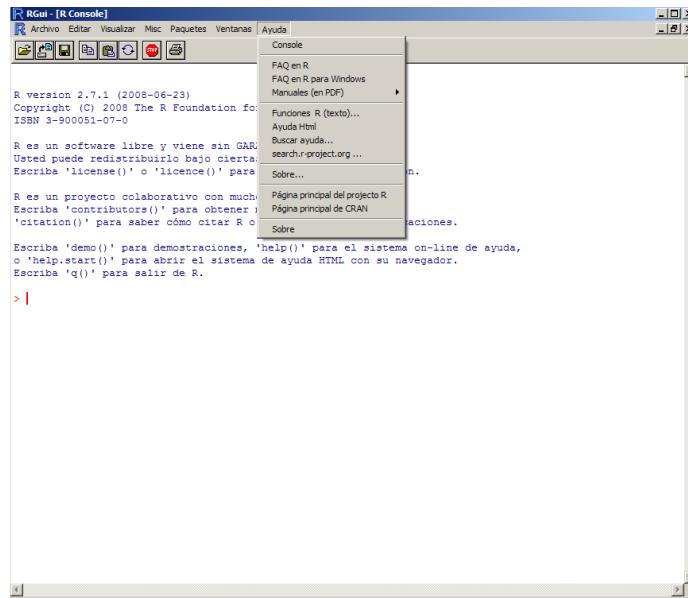
Introducción a R: Inicio sesión

Parte A: Introducción a R



Introducción a R: Ayuda

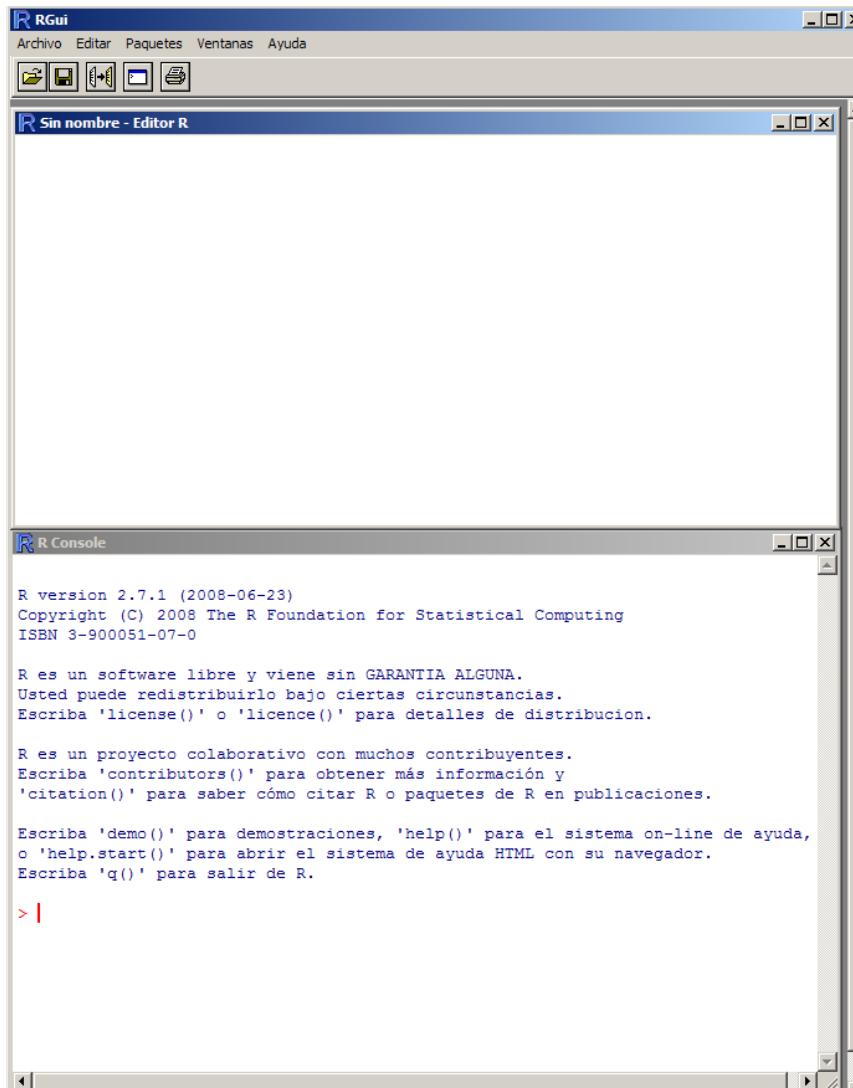
Desde el menu del programa



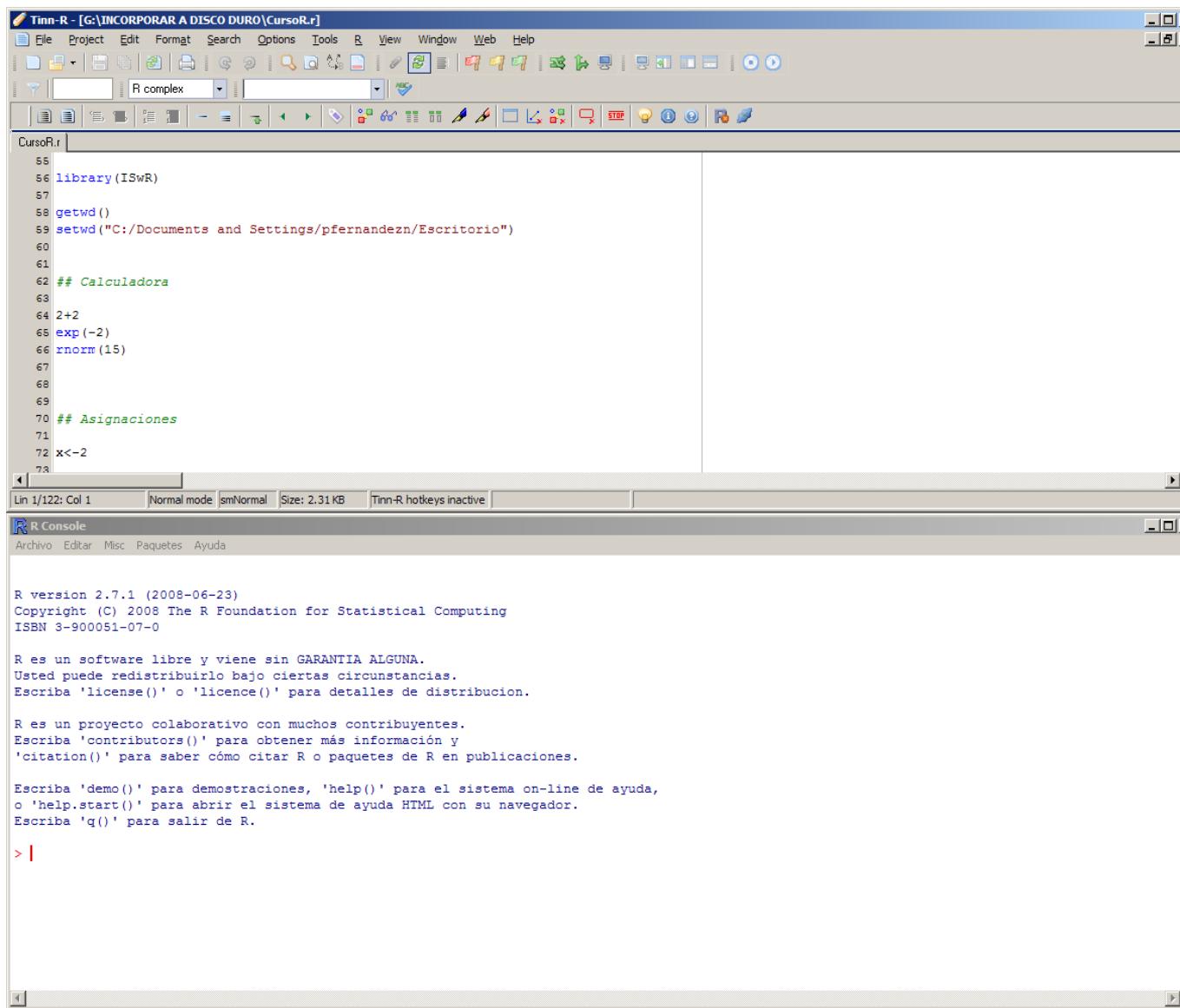
Con comandos

>?rnorm

EDITOR BÁSICO DEL PROPIO R



TinnR



Introducción a R: Editores

Parte A: Introducción a R

The screenshot shows the official TextMate website. At the top, there's a purple header with the TextMate logo and tagline "the missing editor code and markup brought to the 21st century". Below the header is a navigation bar with links for Intro, Blog, Manual, Wiki, and Support. The main content area features a large text block about TextMate's philosophy and a screenshot of the TextMate interface showing a file editor with code and a sidebar. To the right, there are several sections: "Download" (with a download icon), "Purchase" (with a "Buy Now" button), "Documentation" (with a PDF icon), "TextMate in Action" (with a video camera icon), and "Buy the Book!" (with a book icon). A sidebar on the left contains a "Fonts & Colors" configuration panel.

TextMate brings Apple's approach to operating systems into the world of text editors. By bridging UNIX underpinnings and GUI, TextMate cherry-picks the best of both worlds to the benefit of expert scripters and novice users alike.

Whether you are a programmer or a designer, the production of code and markup is hard work. Without an editor dedicated to the task, it is also often cumbersome, overwhelming, and repetitive. Especially when you are dealing with a lot of files at once — like most projects do. TextMate puts you back in control, reduces the mental overhead, and turns manual work into something the computer does.

Created by a closet UNIX geek who was lured to the Mac platform by its ease of use and elegance, TextMate has been referred to as the culmination of Emacs and OS X and has resulted in countless requests for both a Windows and Linux port, but TextMate remains



Introducción a R: Interfaces

Parte A: Introducción a R

The screenshot shows the RStudio desktop application interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Project, Build, Tools, and Help. The status bar at the bottom indicates "10:1" and "(Top Level)".

Code Editor: The left pane displays an R script named "Untitled1.R" with the following code:

```
1 rm(list = ls())
2 N <- 1000
3 u <- rnorm(N)
4 x1 <- -2 + rnorm(N)
5 x2 <- 1 + x1 + rnorm(N)
6 y <- 1 + x1 + x2 + u
7 r1 <- lm(y ~ x1 + x2)
```

Console: The middle-left pane shows the R command history:

```
Tapez <Entrée> pour voir le graphique suivant :
Tapez <Entrée> pour voir le graphique suivant :
Tapez <Entrée> pour voir le graphique suivant :
>
> ?lm
> rm(list = ls())
> N <- 1000
> u <- rnorm(N)
> x1 <- -2 + rnorm(N)
> x2 <- 1 + x1 + rnorm(N)
> y <- 1 + x1 + x2 + u
> r1 <- lm(y ~ x1 + x2)
> |
```

Workspace: The right pane shows the current environment variables:

Values	Type
N	1000
r1	lm[12]
u	numeric[1000]
x1	numeric[1000]
x2	numeric[1000]
y	numeric[1000]

Help Documentation: The bottom right pane displays the "R: Fitting Linear Models" documentation for the `lm` function.

Summary:

Fitting Linear Models

Description

`lm` is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although `aov` may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights,
method = "qr", model = TRUE, x =
singular.ok = TRUE, contrasts =
```

Arguments

<https://posit.co/download/rstudio-desktop/>

PARTE A

Packages

Parte A: Packages

- Tipos de paquetes
- Instalación vs Cargar
- Instalación de paquetes
- Carga de paquetes
- Otras consideraciones

Packages: Tipos de paquetes

A) YA INSTALADOS con el software (Sistema Base)

Ej: stats

B) ADICIONALES: Estos se han de instalar individualmente.

Ej: randomForest, ggplot2

Packages: Instalación vs Cargar

Instalación de paquetes adicionales:

Una vez instalados NO hay que volver a instalarlos.

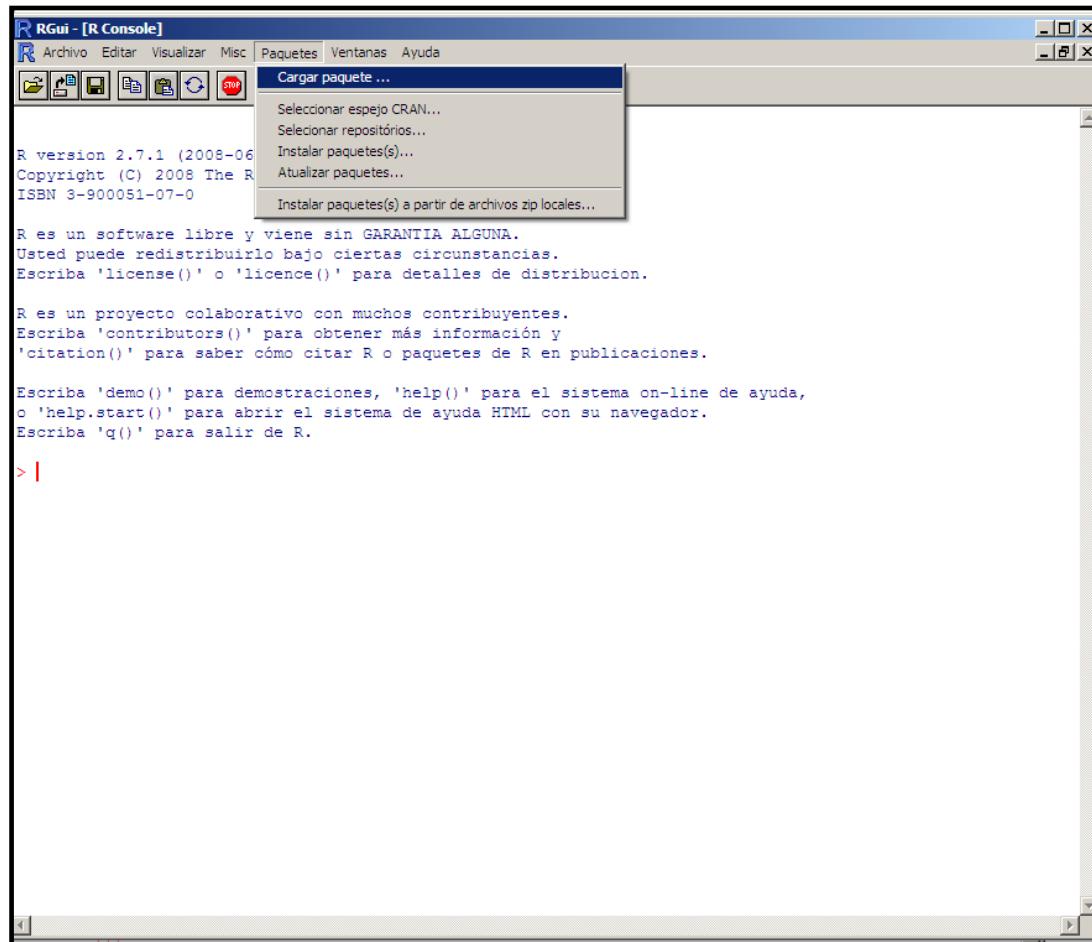
Instalación puede ser a través de menú (Interfaz) o comandos.

Cargar librería:

Uso de funciones de “paquetes adicionales” previamente instalados.

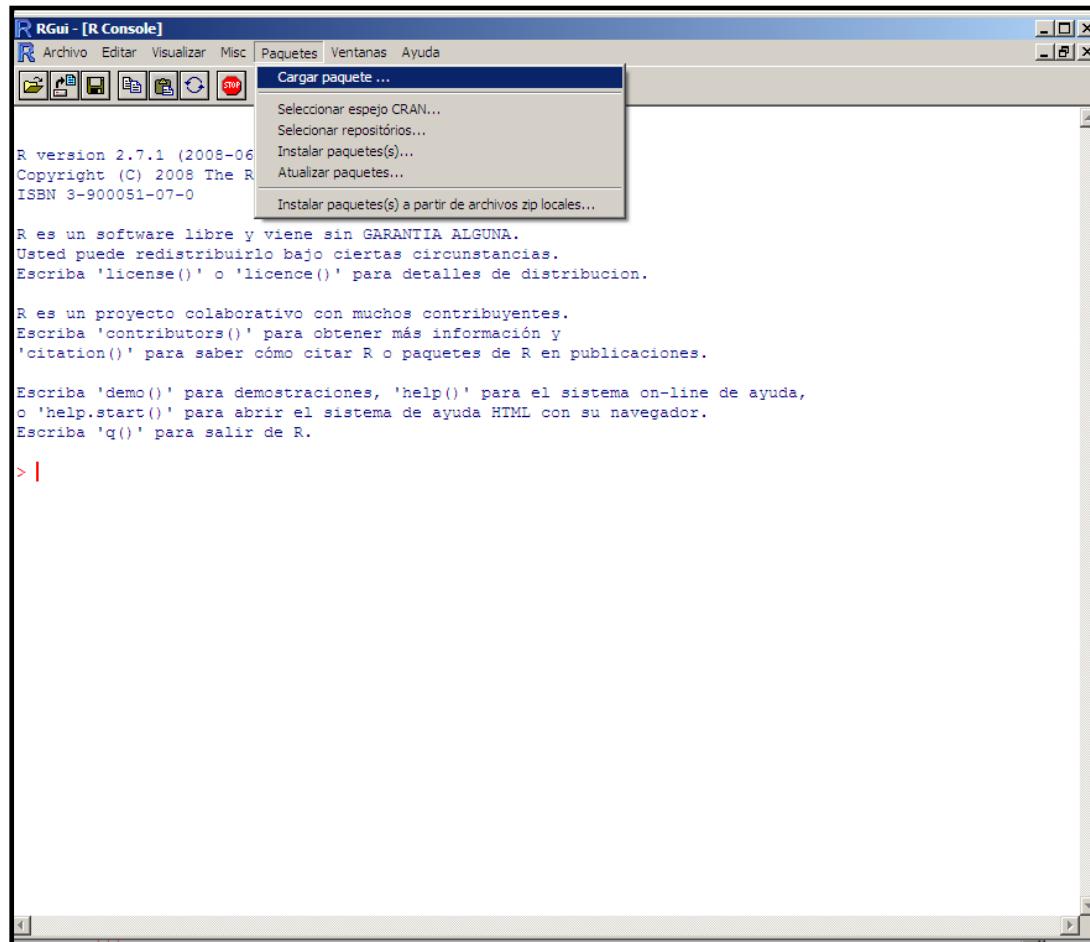
La carga puede ser a través de menú (Interfaz) o comandos.

Packages: Instalación vs Cargar



install.packages()

Packages: Instalación vs Cargar



library()

require()

randomForest

INSTALACIÓN DE PAQUETE ADICIONAL

Packages: Otras consideraciones

Parte A: Packages

- Concepto de librería.
- Actualizaciones: `update.packages()`.
- Creación de paquetes.
- Instalación de paquetes antiguos.
- Instalación de paquetes que no están en repositorio
- Paquetes adicionales almacenados en carpeta de cada versión de R instalada.
- Eliminar paquetes instalados para una versión de R.
- Instalación masiva de paquetes.
- Deshacer la carga de una paquetes en una sesión.
`detach("package:name.of.package", unload=TRUE)`

PARTE A

Importación

Parte A: Importación

- Objetos en R
- Función básica
- Importar formatos especiales
- Importación formato R
- Cargando built-in data

Importación: Objetos en R

- Creación objetos básicos

Tipo de Objeto	Definición	Ejemplo	Observaciones
Vector	Colección ordenada de elementos del mismo tipo	x<-c(1,2,3); z<-c(TRUE,FALSE,TRUE); y<-c("Low", "Low", "Medium", "High")	Factor== tipo de vector de datos cualitativos
Array	Generalización multidimensional del vector. Elementos del mismo tipo.	matrix(rnorm(20),ncol=5)	x
Data.frame	Igual que el array pero puede tener columnas de distintos tipo.	dades <- data.frame(ID=c("gen0", "genB", "genZ"),subj1 = c(10, 25, 33), subj2 = c(NA, 34, 15), oncogen = c(TRUE, TRUE, FALSE),loc = c(1,30, 125))	x
List	Una colección ordenada de objetos conocidos y sus componentes.	una.lista <- c(un.vector = 1:10,una.palabra = "hola",una.matriz = matrix(rnorm(20), ncol = 5),lista2 = c(a = 5,b = factor(c("a", "b"))))	x

- Atributos (nombre, longitud, ...):

class, length, dim, str, is.numeric, is.character, ...

Importación: Objetos en R

WORKSPACE O ÁREA DE TRABAJO



- Espacio de memoria donde se encuentran almacenados los objetos CREADOS o CARGADOS/IMPORTADOS para una sesión
- Espacio de memoria físicamente localizado en Memoria RAM
- Se elimina al cerrar sesión de R

Importación: Objetos en R

- **Ver contenido de un objeto:**

Introducimos el nombre del objeto en la línea de comandos

- **COMANDOS:**

Para saber los objetos que hemos creado o importado/cargado:

> **ls()**

Para borrar objetos concretos

> **rm**(objetos separados por comas)

Para borrar todos los objetos del entorno de trabajo:

> **rm(list = ls())**

Importación: Objetos en R

- Sustitución de nombres de objetos
- Asignación de los valores de un objeto a otro
- No pueden coexistir dos objetos con el mismo nombre (siempre prevalece el más moderno)
- Transformación de clase:

```
Obj1<-as.numeric(Obj1)
```

```
Obj1<-as.character(Obj1)
```

```
Obj1<-as.numeric(as.character(Obj1))
```

Importación: Función básica

read.table ()

```
?read.table
```

```
datos <- read.table(file="mis_datos.txt", header = FALSE, sep = "|")
```

Importación: Importar formatos especiales

read.csv

read.xls == package **gdata**

read.xlsx==package **xlsx**

read.spss == package **foreign**

read.dta == package **foreign**

read.dta13 == package **readstata13**

read.fwf

Importación: Importar formato R (workspace)

- Vamos a hablar de cargar
- Formato: Rdata o RData

load ()

```
> load(file="un_workspace_guardado.RData")
```

Importación: Cargando built-in data

R, y muchos paquetes, incorporan ficheros con datos.

```
> library(multtest)
```

```
> ?golub
```

```
> data(golub)
```

```
> data(golub, package = "multtest")
```

PARTE A

Manejo básico de datos

Data.frame

Parte A: MBD data.frame

- Data frame
- Operaciones con variables (vectores)
- Generación de secuencias
- Missing
- Ordenación

datos.curso1.RData

'data.frame': 200 obs. of 11 variables:

```
$ ID      : num  137 174 200 23 39 90 40 115 72 27 ...
$ edad    : num  37 85 29 13 49 12 85 31 39 70 ...
$ sexo    : chr "Mujer" "Mujer" "Hombre" "Hombre" ...
$ estado.civil : chr "Casado" "Soltero" "Casado" "Divorciado" ...
$ nivel.estudios : chr "Bajo" "Alto" "Bajo" "Alto" ...
$ peso     : num  59.6 60 79.2 80.8 80.8 ...
$ altura   : num  151 149 169 171 171 ...
$ fumador  : chr "No" "No" "No" "Si" ...
$ diabetes : chr "No" "Si" "Si" "Si" ...
$ cancer.mama : chr "Si" "No" "Si" NA ...
$ cancer.prostata: chr NA NA "Si" "Si" ...
```

	ID	edad	sexo	estado.civil	nivel.estudios	peso	altura	fumador	diabetes	cancer.mama	cancer.prostata
1	137	37	Mujer	Casado	Bajo	59.58221	150.7183	No	No	Si	<NA>
2	174	85	Mujer	Soltero	Alto	59.95427	149.2075	No	Si	No	<NA>
3	200	29	Hombre	Casado	Bajo	79.20674	168.9795	No	Si	Si	Si
4	23	13	Hombre	Divorciado	Alto	80.78347	171.1568	Si	Si	<NA>	Si
5	39	49	Hombre	Divorciado	Bajo	80.76036	170.5682	Si	Si	<NA>	No
6	90	12	Hombre	Casado	Alto	79.83426	170.8565	No	No	<NA>	Si
7	40	85	Hombre	Casado	Alto	80.69636	168.5586	No	No	<NA>	Si
8	115	31	Mujer	Soltero	Alto	61.28985	150.0667	Si	No	Si	<NA>
9	72	39	Mujer	Divorciado	Bajo	60.70871	150.4091	Si	Si	Si	<NA>
10	27	70	Hombre	Soltero	Medio	78.89874	167.8307	No	Si	<NA>	Si
11	19	24	Mujer	Divorciado	Alto	60.06984	149.5328	No	No	Si	<NA>
12	133	45	Hombre	Soltero	Medio	79.57263	170.7013	Si	Si	<NA>	Si
13	15	42	Mujer	Soltero	Bajo	60.39387	150.7226	Si	No	Si	<NA>
14	44	74	Mujer	Casado	Medio	60.33449	148.2137	No	Si	Si	<NA>
15	179	16	Mujer	Divorciado	Medio	60.01237	151.4407	Si	Si	Si	<NA>
16	148	6	Mujer	Soltero	Bajo	57.93049	149.7306	No	No	Si	<NA>
17	192	31	Mujer	Casado	Bajo	57.99527	148.3326	Si	Si	Si	<NA>

MBD data.frame: Data.frame

- **Creación vs Importación / cargar**
- **Visualización**
- **Acceso a elementos**
- **Recodificación elementos**
- **Añadir variables / columnas**
- **Añadir registro / filas**
- **Eliminar columnas / filas**
- **Nombres variables / indicadores de fila**

MBD data.frame: Data.frame

- **Creación:**

```
> my.data.frame <- data.frame(ID = c("paciente1", "paciente2", "paciente3"),  
    edad= c(10, 25, 33), altura = c(NA, 150, 180),  
    cancer = c(TRUE, TRUE, FALSE),  
    bmi = c(22,25, 20))
```

- **Importación / cargar:**

```
> load("datos.curso1.RData")
```

MBD data.frame: Data.frame

- Visualización:

```
> head(datos,3)
```

```
> tail(datos,3)
```

```
> head(datos,3)
  ID edad sexo estado.civil nivel.estudios peso altura fumador diabetes cancer.mama cancer.prostata
1 137   37 Mujer     Casado      Bajo 59.58221 150.7163    No     No      Si      <NA>
2 174   85 Mujer     Soltero    Alto 59.95427 149.2075    No     Si      No      <NA>
3 200   29 Hombre   Casado      Bajo 79.20674 168.9795    No     Si      Si      Si
> tail(datos,3)
  ID edad sexo estado.civil nivel.estudios peso altura fumador diabetes cancer.mama cancer.prostata
198 57   41 Hombre   Soltero    Medio 79.45077 168.4157    No     No      <NA>      No
199 169  44 Mujer     Soltero    Bajo 59.69171 149.0113    Si     No      No      <NA>
200 143  54 Mujer   Divorciado Medio 61.02373 150.8797    Si     Si      Si      <NA>
```

```
> fix(datos)
```

```
> str(datos)
```

```
> str(datos)
'data.frame': 200 obs. of 11 variables:
 $ ID          : num 137 174 200 23 39 90 40 115 72 27 ...
 $ edad        : num 37 85 29 13 49 12 85 31 39 70 ...
 $ sexo         : chr "Mujer" "Mujer" "Hombre" "Hombre" ...
 $ estado.civil: chr "Casado" "Soltero" "Casado" "Divorciado" ...
 $ nivel.estudios: chr "Bajo" "Alto" "Bajo" "Alto" ...
 $ peso         : num 59.6 60 79.2 80.8 80.8 ...
 $ altura       : num 151 149 169 171 171 ...
 $ fumador      : chr "No" "No" "No" "Si" ...
 $ diabetes     : chr "No" "Si" "Si" "Si" ...
 $ cancer.mama : chr "Si" "No" "Si" NA ...
 $ cancer.prostata: chr NA NA "Si" "Si" ...
```

- **Acceso a elementos:**

a) A una variable: nombre.data.frame\$"nombre .variable"

```
> datos$"sexo"  
> sexo.mirar <- datos$"sexo"
```

b) Notacion matricial (Nombre.data.frame[Filas, Columnas])

```
> datos[,3]; datos[1:3,3]; datos [1:10,1:3]
```

c) Vector de indices(posiciones), vector lógico, condición lógica sobre otros vectores (variables), etc

```
> datos [datos$"sexo"=="Mujer", ]
```

```
> datos [datos$"sexo"=="Mujer" & datos$"estado.civil"=="Casado" ,1:10 ]
```

```
> datos [datos$"sexo">%in%“Mujer” & datos$"estado.civil">%in%“Casado” ,1:10 ]
```

```
> datos.new <- datos [datos$"sexo"=="Mujer", ]
```

- **Recodificación:**

```
> datos.new2<-datos
```

```
> datos.new2[1 , 3] <- "Hombre"
```

```
> datos.new2[datos.new2$"ID"==200 , sexo ]<-"Mujer"
```

```
> datos.new2$"sexo"[datos.new2$"ID"==200]<-"Mujer"
```

MBD data.frame: Data.frame

- **Añadir variables (columnas):**

a) nombre.data.frame\$"nombre.nueva.variable" <- valores

```
> datos$"caso.cancer" <- c(1,1,1.....)
```

b) Utilizar la funcion cbind()

```
> datos.nuevos1 <- datos[ ,1:3]
```

```
> datos.union1 <- cbind(datos,datos.nuevos1)
```

c) Utilizar la funcion merge() (YA LA VEREMOS)

MBD data.frame: Data.frame

- **Añadir registros (filas):**

- a) Utilizar la funcion rbind()

```
> datos.nuevos2 <- datos[1:5, ]
```

```
> datos.union2 <- rbind(datos,datos.nuevos2)
```

- **Eliminar variables o registros (1/2):**

- a) nombre.data.frame [-vector de posiciones de las filas a eliminar, - vector de posiciones de las columnas a eliminar]

```
> datos.nuevos3 <- datos[-c(2,3), ]
```

```
> datos.nuevos4 <- datos[-c(2,3) , -c(14,18)]
```

MBD data.frame: Data.frame

- **Eliminar variables o registros (2/2):**

b) nombre.data.frame [! condiciones lógicas, ! Condiciones logicas]

```
> indice5 <- datos$"sexo"!="Mujer"
> datos.nuevos5 <- datos[indice5, ]

> datos.nuevos6 <- datos[ , names(datos)!="peso"]

> indice7 <- which(datos$"sexo"=="Mujer")
> datos.nuevos7 <- datos[-indice7, ]
```

MBD data.frame: Data.frame

- Nombres variables / indicadores de fila

> **names(datos)**

```
[1] "ID"      "edad"     "sexo"      "estado.civil"  "nivel.estudios" "peso"  
[7] "altura"   "fumador"   "diabetes"   "cancer.mama"   "cancer.prostata"
```

> **names(datos)[3]**

```
[1] "sexo"
```

> **names(datos)[3]<- "sex"**

> **nombres.variables<-names(datos)**

> **row.names(datos)**

```
[1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16" ...
```

> **mujeres<-datos[datos\$sexo%in%"Mujer",]**

> **row.names(mujeres)**

```
[1] "1"  "2"  "8"  "9"  "11" "13" "14" "15" "16" "17"...
```

> **row.names(mujeres)<-NULL**

> **row.names(mujeres)**

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

MBD data.frame: Operaciones con variables

- **Operaciones aritméticas**
- **Operaciones comparativas / lógicas**
- **Operaciones con conjuntos**

MBD data.frame: Operaciones con variables

- **Operaciones aritméticas**

- suma `+`, resta `-`, multiplicación `*`, división `/`

- potencia `^`, raíz cuadrada `sqrt`

- `%/%` división entera, `%%` módulo: resto de la división entera

- logaritmos `log`, `log10`, `log2`, `logb(x, base)`, exponencial `exp`

- trigonométricas `sin`, `cos`, `tan`, `asin`, `acos`, `atan`

- otras:

- `max`, `min`, `range`, `pmax`, `pmin`, `mean`, `median`, `var`, `sd`, `quantile`.

- `sum`, `prod`, `diff`, `cumsum`, `cumprod`, `cummax`, `cummin`.

MBD data.frame: Operaciones con variables

- **Operaciones aritméticas**

```
> datos$ratio <- datos$peso/datos$altura
```

```
> datos$edad_new <- datos$edad/2
```

```
> datos$edad_2 <- datos$edad + datos$edad_new
```

MBD data.frame: Operaciones con variables

- **Operaciones comparativas / lógicas**

<, >, <=, >=, ==, !=

!, &, |, xor() y los parecidos &&, ||

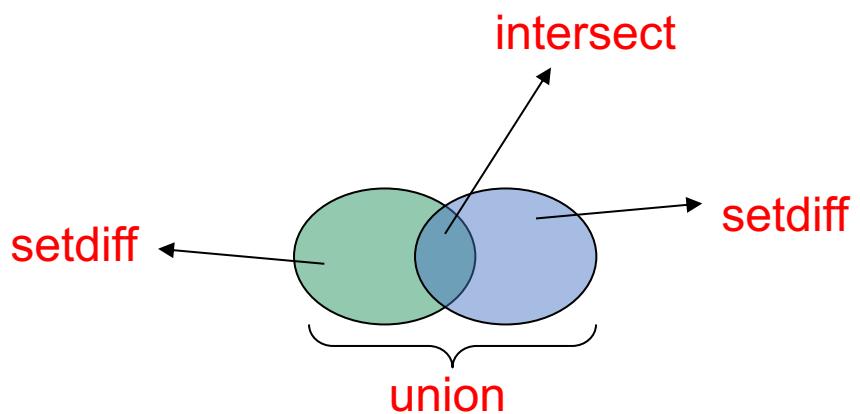
```
> datos$check_edad <- datos$edad>40  
> table (datos$check_edad,exclude=NULL)
```

```
> datos$check_edad_2 <- datos$edad==30  
> table (datos$check_edad_2,exclude=NULL)
```

MBD data.frame: Operaciones con variables

- **Operaciones con conjuntos**

```
> x <- 1:5; y <- c(1, 3, 7:10)  
> union(x, y)  
> intersect(x, y)  
> setdiff(y, x)  
  
> v <- c("bcA1", "bcA2", "blX1")  
> w <- c("bcA2", "xA3")  
> union(v, w)  
> intersect(v, w)  
> setdiff(w, v)  
> setdiff(v, w)
```



MBD data.frame: Generación de secuencias

- **Secuencias aleatorias**

NO basadas en una distribución de probabilidad definida

```
> sample()
```

Basadas en una distribución de probabilidad definida

```
> rnorm(10)
```

```
> rnorm(10, mean = 13, sd = 18)
```

- **Secuencias NO aleatorias**

```
> datos$ID_new1<- seq(from = 1, to = dim(datos)[1], by = 1)
```

```
> datos$ID_new2<- rep(1, dim(datos)[1])
```

MBD data.frame: Missing

- **NA**

```
> sum(is.na(datos$edad))
```

```
> which(is.na(datos$edad))
```

```
> datos$edad[is.na(datos$edad)]<- 10
```

```
> datos$missing<-is.na(datos$edad)
```

MBD data.frame: Ordenación

- **sort / order**

```
> datos_new <- datos[order(datos$edad),]
```

```
> datos_new2 <- datos[order(datos$edad,datos$peso),]
```

```
> datos_new3 <- datos[order(datos$edad,decreasing=T),]
```

PARTE A

Manejo básico de datos

Fechas y caracteres

MBD fechas

Clase => “Date”

as.Date()

> date()

[1] "Sat May 14 21:01:58 2016"

Internamente las fechas son representadas como el número de días después o antes de un punto conocido como “epoch” (January 1, 1970)

MBD fechas

- **Formato fecha perfecto para R: 1977-05-05**

```
class(datos$"fdiag_cm")
sum(is.na(datos$"fdiag_cm"))
unique(datos$"fdiag_cm")
table(datos$"fdiag_cm")
```

```
datos$"fechas.CM" <- as.Date(datos$"fdiag_cm")
```

```
class(datos$"fechas.CM")
sum(is.na(datos$"fechas.CM"))
datos$"fechas.CM"[1:6]
head(datos[,c("fdiag_cm","fechas.CM")])
```

MBD fechas

- Otro formato de fecha: **10.07.88**

```
class(datos$"fdiag_cp")
sum(is.na(datos$"fdiag_cp"))
unique(datos$"fdiag_cp")
```

```
datos$"fechas.CP" <- as.Date(datos$"fdiag_cp",format="%d.%m.%y")
```

?strptime

```
head(datos[,c("fdiag_cp","fechas.CP")])
```

```
class(datos$"fechas.CP")
sum(is.na(datos$"fechas.CP"))
unique(datos$"fechas.CP")
```

MBD operaciones con fechas

```
datos$"dias"><-c(datos$"fechas.DF" - datos$"fechas.CM")  
class(datos$"dias")
```

```
datos$"dias2"<-difftime(datos$"fechas.DF", datos$"fechas.CM",units="days")
```

```
datos$"weeks"<-difftime(datos$"fechas.DF", datos$"fechas.CM",units="weeks")
```

```
datos$"years"<-difftime(datos$"fechas.DF", datos$"fechas.CM",units="days") / 365.25  
attr(datos$"years","units")="years"
```

MBD secuencias fechas

```
datos_secuencia<-data.frame(fecha=seq(as.Date("1976-01-01"),to=as.Date("1976-03-01"),by="days"))
```

```
datos_infeccion<-data.frame(fecha=c("1976-02-01","1976-02-01","1976-02-01","1976-03-01","1976-03-01"),
infeccion=rep("Si",5))
table(datos_infeccion$fecha)
```

```
seq(as.Date("1976-01-01"),by="days",length=6)
```

```
seq(as.Date("1976-01-01"),to=as.Date("1976-03-01"),by="days")[1:6]
```

```
seq(as.Date("1976-01-01"),to=as.Date("1976-03-01"),by="2 weeks")[1:6]
```

```
seq(as.Date("1976-01-01"),to=as.Date("1976-03-01"),by="week")[1:6]
```

MBD caracteres

nchar() # numero de caracteres (de la librería “gdata”)

paste() # concatenar caracteres

strsplit() # Dividir los elementos de un vector de caracteres en subcadenas

substring() # Extraer o reemplazar subcadenas en un vector de caracteres.

toupper, tolower() # mayúsculas y minúsculas

sub(), gsub() # buscar coincidencias y reemplazar dentro de un vector de caracteres

unique() # lista los elementos de un vector sin duplicados

duplicated()

grep y match

MBD caracteres

- **nchar() # numero de caracteres (de la librería gdata)**

```
nchar(as.character(datos$"ID"))
table(nchar(as.character(datos$"ID")))
```

- **paste() # concatenar caracteres**

```
datos$"id_combinado"<-paste(datos$"ID",datos$"sexo",sep="***")
head(datos[,c("ID","sexo","id_combinado")])
```

```
x <- c("asfef", "qwerty", "yuiop[", "b", "stuff.blah.yech")
paste(x,collapse="/")
```

```
datos$"ID_new"<-datos$"ID"
datos$"ID_new"[nchar(datos$"ID")==1]<-
paste("00",datos$"ID_new"[nchar(datos$"ID")==1],sep="")
datos$"ID_new"[nchar(datos$"ID")==2]<- paste("0",datos$"ID_new"[nchar(datos$"ID")==2],sep="")
```

MBD caracteres

- **strsplit() # Dividir los elementos de un vector de caracteres en subcadenas**

```
res<-strsplit(datos$"fdiag_cm",split="-") # el resultado es un objeto tipo lista
```

```
datos$"fdiag_cm"[1]  
res[[1]]
```

```
res<-do.call(rbind.data.frame, res)  
names(res)<-c("year","month","day")
```

```
table(res$"year",exclude=NULL)
```

```
datos$"year"<-res$"year"  
datos$"month"<-res$"month"  
datos$"day"<-res$"day"
```

```
head(datos[,c("ID","fdiag_cm","year","month","day")])
```

MBD caracteres

- **substring**

```
table(nchar(as.character(datos$"ID")))
```

```
datos$ID_new<-substring(as.character(datos$"ID"),first=1,last=2)
```

```
datos$"month_new"<-substring(datos$"fdiag_cm",first=6,last=7)
```

- **toupper / tolower**

```
nombres<-c("María","Pachón")
```

```
nombres<-tolower(nombres)
```

```
nombres<-toupper(nombres)
```

MBD caracteres

- **sub/gsub**

```
sub("á","a",c("hólálá"))
```

```
nombres<-gsub("á","a",nombres)  
nombres<-gsub("é","e",nombres)
```

```
nombres<-gsub("ñ","n",nombres)
```

```
nombres<-gsub("ü","u",nombres)  
nombres<-gsub("ö","o",nombres)
```

- **unique**

```
unique(c("AA","BB","AA","AA","CC","DD","DD"))
```

MBD caracteres

- **duplicated**

```
which(duplicated(c("AA","BB","AA","AA","CC","DD","DD")))
```

```
datos$"ID"[4]<-200  
head(datos)
```

```
table(duplicated(datos$"ID"))  
length(datos$"ID")  
length(unique(datos$"ID"))
```

```
datos$"duplicado_id"<-duplicated(datos$"ID")
```

```
datos[datos$"ID">%in%datos$"ID"[datos$"duplicado_id"],]
```

Eliminar los duplicados

```
datos$"duplicado_perfecto"<-duplicated(paste(datos$"ID",datos$"edad",datos$"sexo",sep=""))  
table(datos$"duplicado_perfecto")
```

```
datos_new<-datos[-which(datos$"duplicado_perfecto"), ]
```

- **match y grep**

```
nombres<-c("Enrique","Susana","Lidia","Estefanía","Enrique","EnriqueIV","enrique","Estefania")
```

```
match("Enrique",nombres)
```

```
grep("Enrique",nombres)
```

```
datos$$observaciones<-"cardiovascular"
```

```
datos$$observaciones[c(10,15,16)]<-"cardiovascular;cancer"
```

```
indice_cancer<-grep("cancer",datos$$observaciones")
```

```
datos$cancer<-0
```

```
datos$cancer[indice_cancer]<-1
```

```
datos$$infeccion<-"No"
```

```
datos$$infeccion[c(2,5,6)]<-"infección por rinovirus"
```

```
datos$$infeccion[c(8,9,10)]<-"infección por rhinovirus"
```

```
busqueda1<-grep("rinovirus",datos$$infeccion")
```

```
busqueda2<-grep("rhinovirus",datos$$infeccion")
```

```
busqueda_total<-unique(c(busqueda1,busqueda2))
```

```
datos$rinovirus<-0
```

```
datos$rinovirus[busqueda_total]<-1
```

PARTE A

Manejo básico de datos

Combinación y reestructuración

MBD Combinación y reestructuración

- **Combinación de bases de datos (rbind)**
- **Estratificación de base de datos (split)**
- **Cruzando bases de datos (merge)**
- **Reestructuración de bases de datos (reshape)**

MBD Combinación

- **rbind**

```
require(data.table)
```

```
DF1 <- data.table(id="Mario", altura=172, color.ojos="azules")
```

```
DF2<- data.table(id="Pablo", altura=202, color.ojos="amarillos")
```

```
DF3 <- data.table(id="Lucía", altura=155, color.ojos="negros")
```

```
rbind(DF1,DF2,DF3)
```

MBD Estratificación

- **subset**

```
datos <- data.table(datos)
```

```
datos.hombres<- subset(datos, sexo=="Hombre", select=c(sexo,peso))
```

```
datos.mujeres <- subset(datos, sexo=="Mujer", select=c(sexo,peso))
```

- **subset & split**

```
temp <- subset(datos,select=c(sexo,estado.civil,peso))
```

```
estratos1 <- split(temp, by="sexo")
```

```
estratos2 <- split(temp, by=c("sexo","estado.civil")) )
```

MBD Cruce

- **merge**

```
DF1=data.table(id=c("Mario", "Pablo", "Lucía"),altura=c(167,96,228))
```

```
DF2=data.table(id=c("Mario", "Pablo"), peso=c(75,32))
```

```
merge(DF1,DF2,by="id",all.x=T)
```

MBD Reestructuración

- **Formato wide: una columna para cada variación**

```
Wide <- subset(datos, sexo=="Mujer", select=c(ID:sexo,fdiag_cm,fdef))
head(Wide)
```

- **Formato long: un registro para cada variación**

```
Long <- melt(Wide, id=1:3) # id : variables que se quedan fijas
setkey(Long,ID) # reordenando la base por ID
head(Long)
```

- **Volviendo al formato wide**

```
Wide_new <- dcast(Long, ID + sexo + edad ~ variable)
head(Wide_new)
```