

The logo for the SGG Virtual Conference 6th Edition. It features the letters 'SGG' in a large, bold, green font. To the right of 'SGG' is a green globe icon. Below 'SGG' and the globe, the word 'VIRTUAL' is written in a smaller, green, sans-serif font. Below 'VIRTUAL', the word 'CONFERENCE' is written in a larger, bold, green, sans-serif font. At the bottom of the logo, the text '6ta edición' is written in a green, sans-serif font.

SGG
VIRTUAL
CONFERENCE
6ta edición

Introducción a R para Desarrolladores

Presentado por:
Alan Lazalde

AGENDA

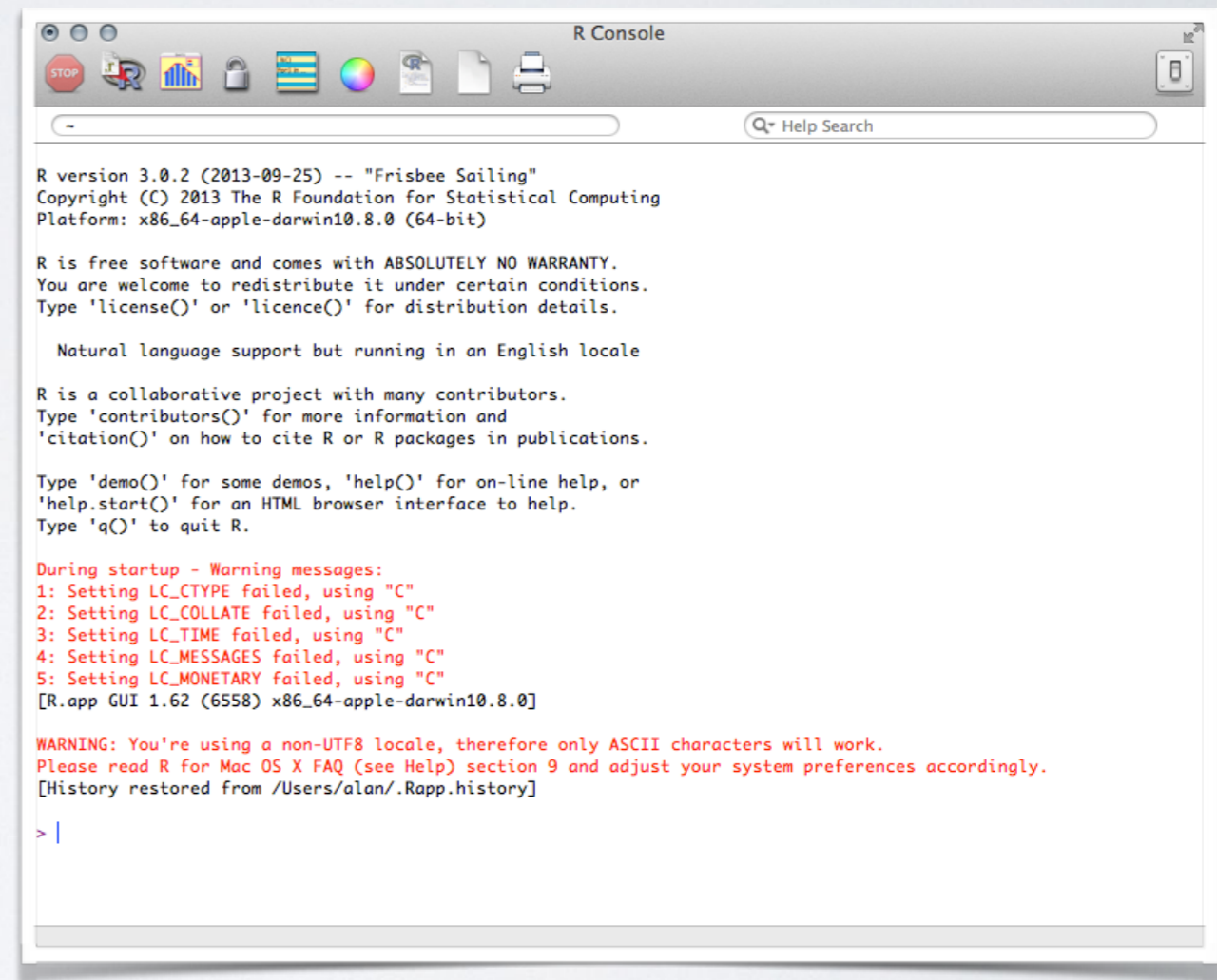
- Qué es R y cómo surge
- Usuarios y usos
- Qué instalar
- El lenguaje...
- Y sus peculiaridades



QUÉ ES

1. Un lenguaje de programación

2. El software libre **multiplataforma** que ejecuta programas escritos en R y es altamente extensible



```
R Console
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (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.

During startup - Warning messages:
1: Setting LC_CTYPE failed, using "C"
2: Setting LC_COLLATE failed, using "C"
3: Setting LC_TIME failed, using "C"
4: Setting LC_MESSAGES failed, using "C"
5: Setting LC_MONETARY failed, using "C"
[R.app GUI 1.62 (6558) x86_64-apple-darwin10.8.0]

WARNING: You're using a non-UTF8 locale, therefore only ASCII characters will work.
Please read R for Mac OS X FAQ (see Help) section 9 and adjust your system preferences accordingly.
[History restored from /Users/alan/.Rapp.history]

> |
```

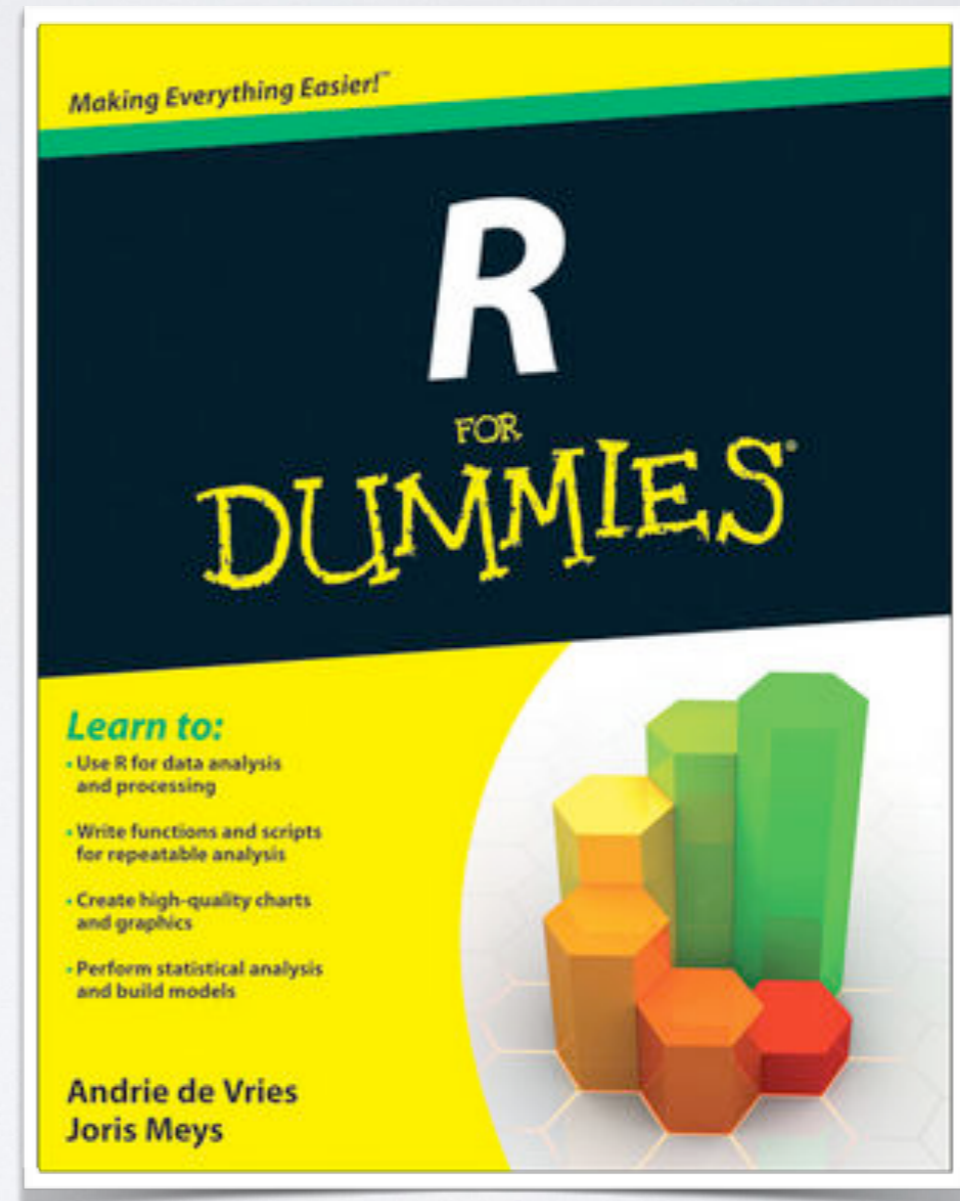
EL ORIGEN

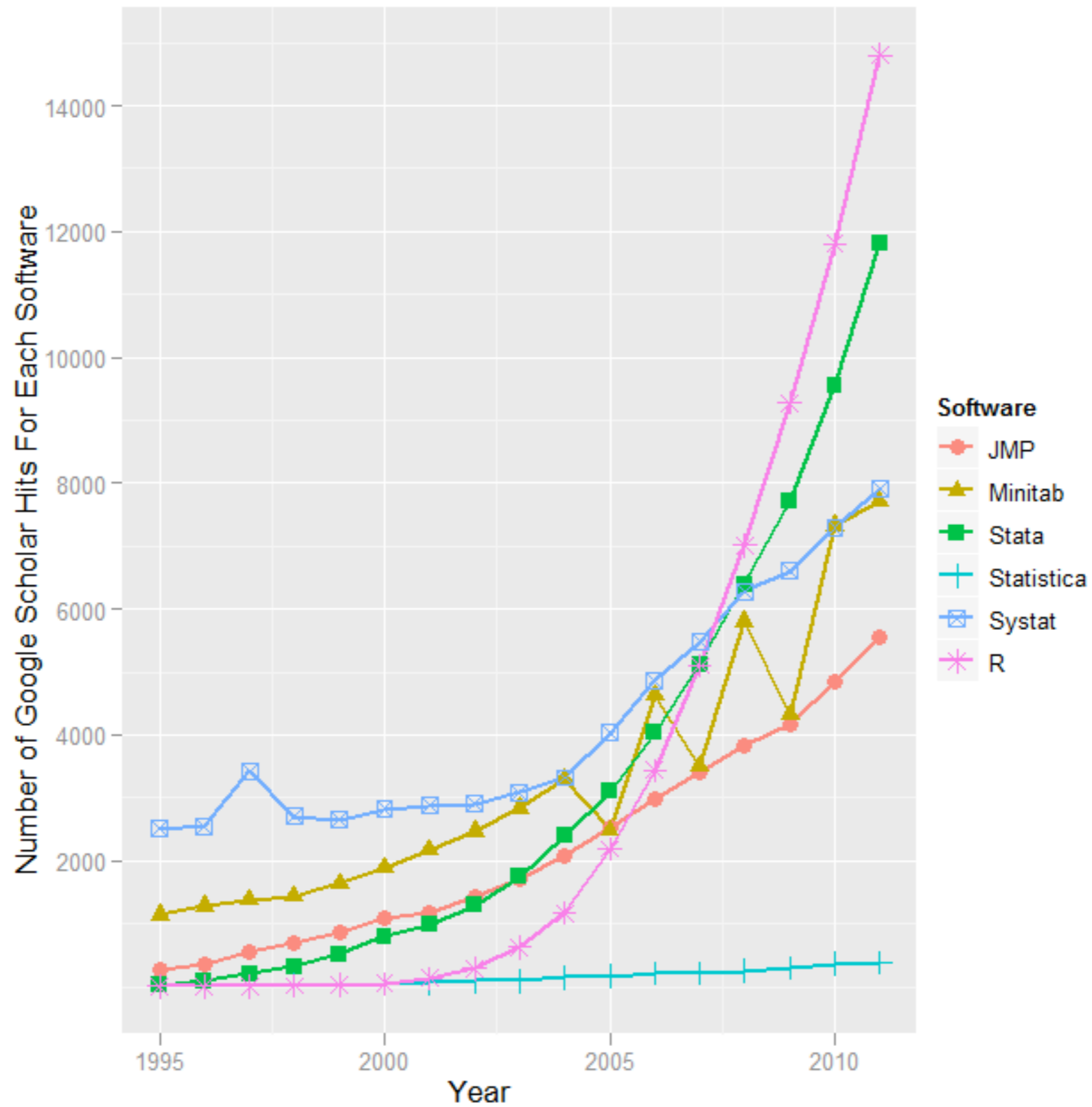
- Ross Ihaka (izq.) y Robert Gentleman (der.), 1993
- Es un dialecto del lenguaje S de Bell Systems (1970's)
- Actualmente hay unas 20 personas trabajando en R



QUIÉNES LOS USAN

- Matemáticos
- Biólogos
- Científicos de datos
- Financieros
- Y cada vez más programadores...





POPULARIDAD

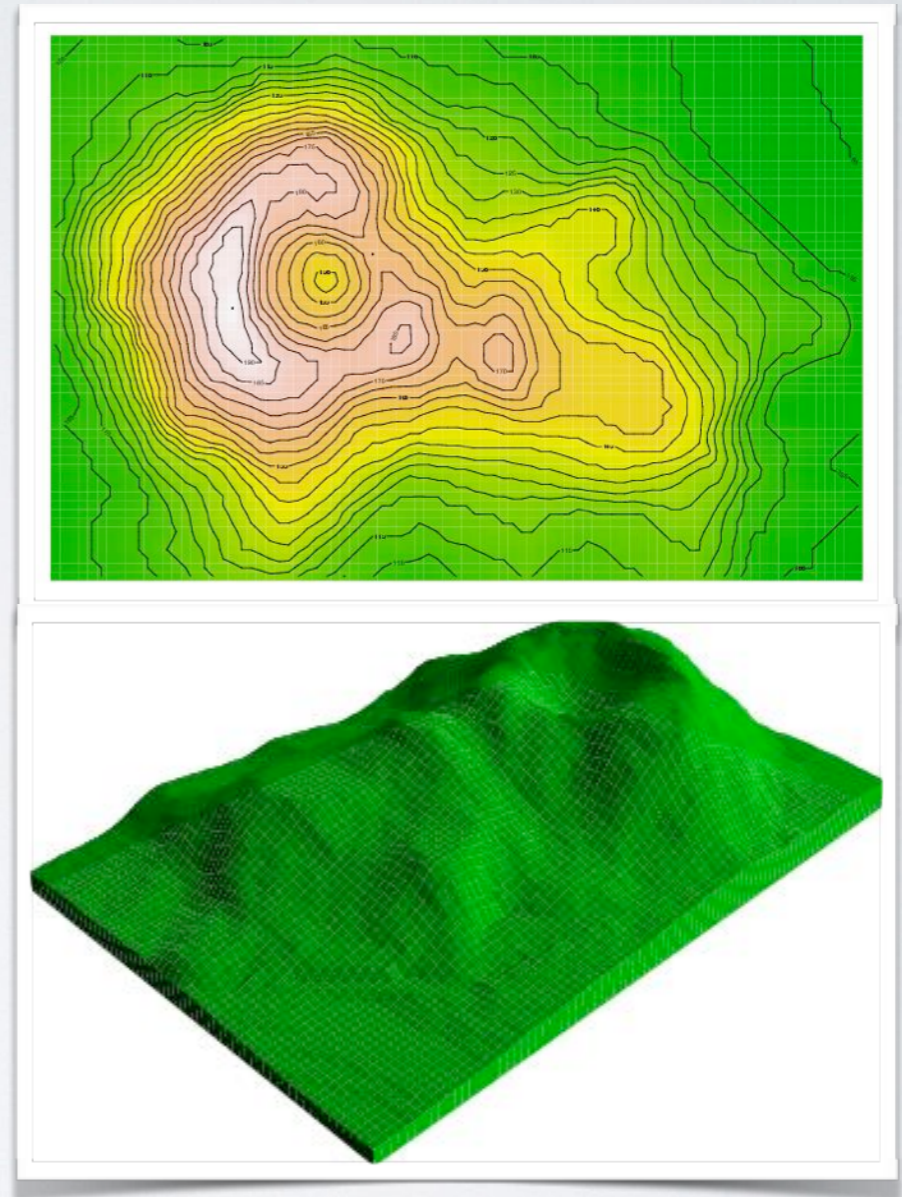
EMPRESAS



... *R* is used in the news cycle at the **The New York Times** to crunch data and prepare graphics before they go to print or online..., beyond "Here is some data:" toward something closer to inference."

PARA QUÉ LO USAN

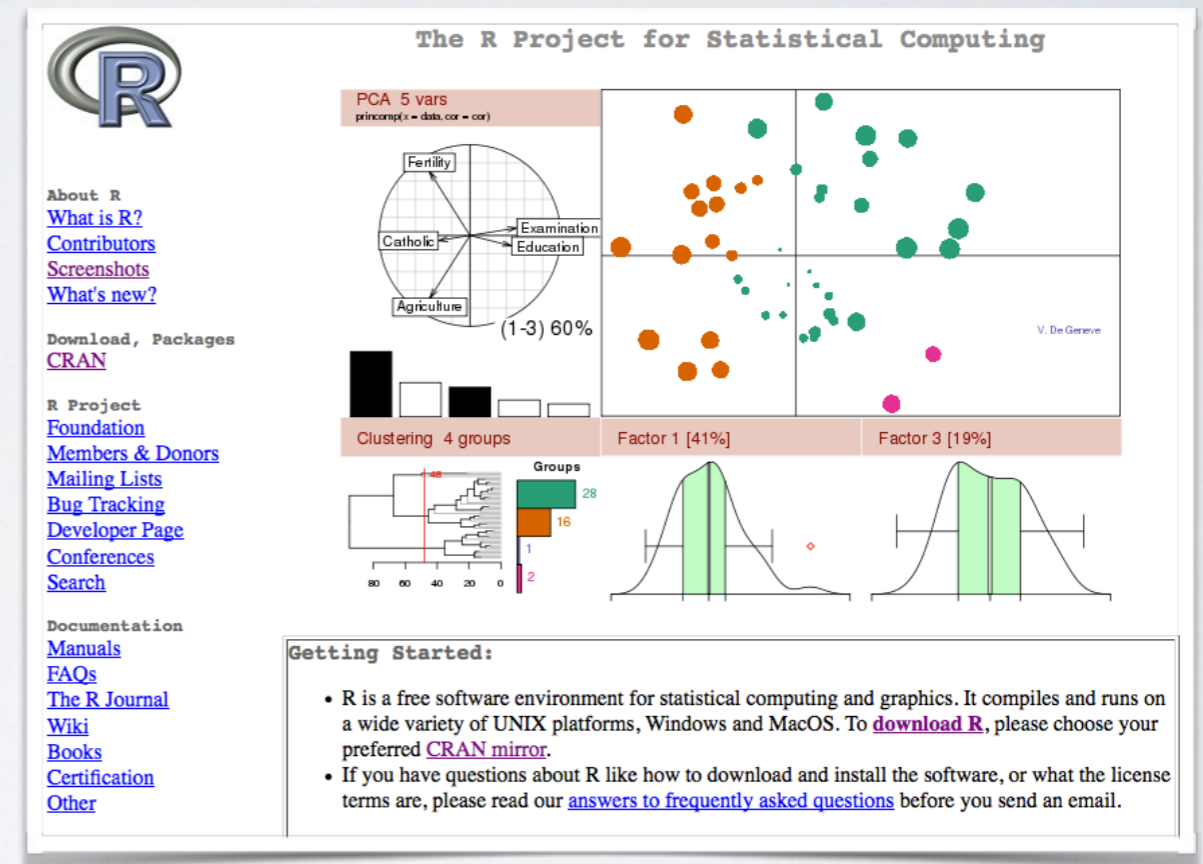
- Estadística
- Visualización
- Supercómputo
- Finanzas
- Inteligencia artificial (*machine learning*)
- Imagenología médica
- Econometría
- Física computacional



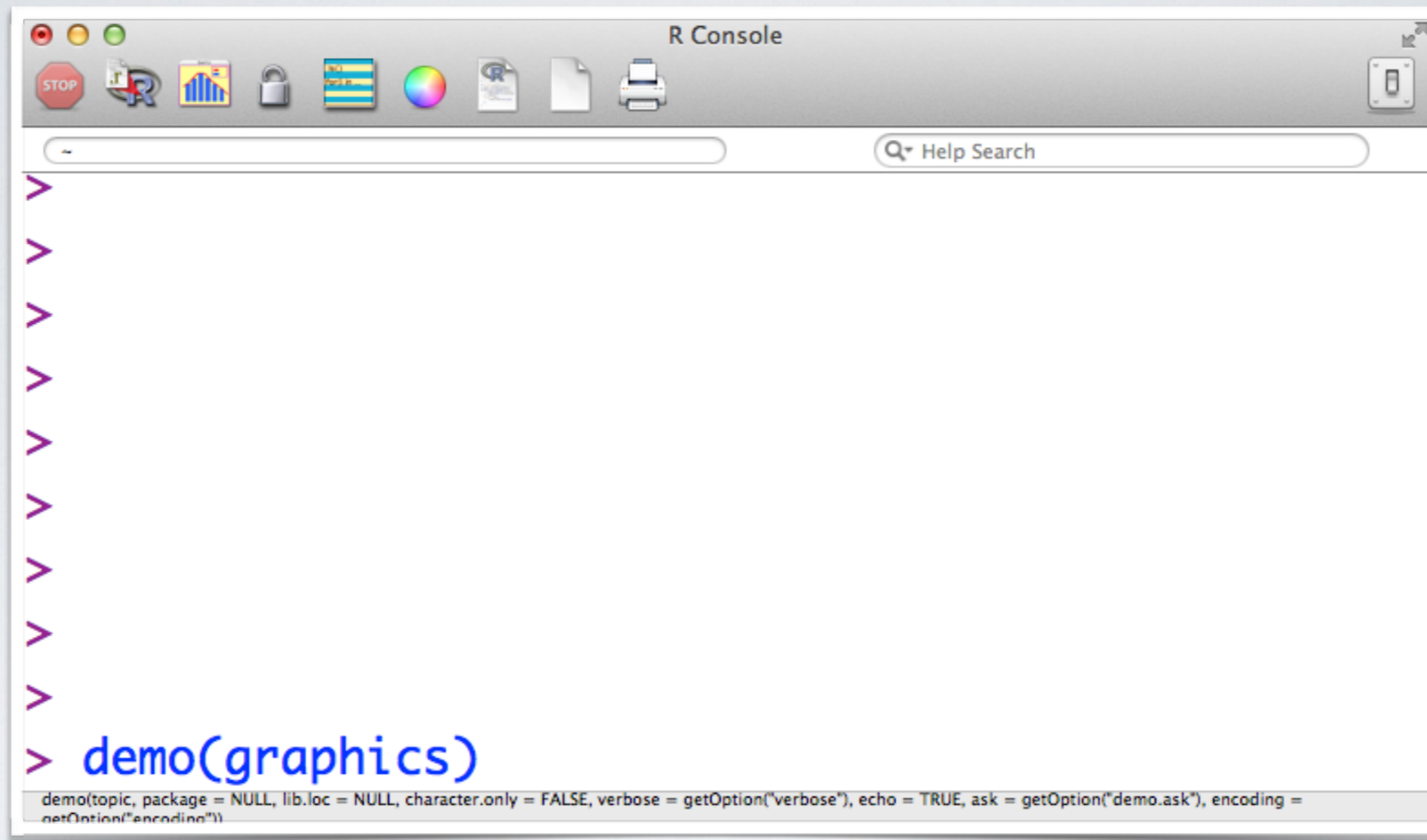
INSTALACIÓN

INSTALACIÓN

1. Ir a <http://cran.r-project.org/mirrors.html>
2. Elegir un mirror
3. Elegir una versión para tu sistema operativo (Linux, Windows, OS X; código fuente y binarios)



¿Dudas? <http://cran.r-project.org/doc/manuals/R-admin.html>



R CONSOLE

El IDE básico

IDE'S

1. RStudio
2. RExcel
3. Rguis
4. Vim-R-Tmux
5. Emacs (ESS)
6. Tinn-R
7. RKWard
8. RPy

```
4+ script2.R zzz.R RNA-Seq.R X
| -fmsR-manual | x, Rowv = NULL, Colv = if (symm) "Rowv" else NULL, distfun, hclust
| -fmsR-manual |
| -fmsR-manual |
| ~Rdlatex.log | [Scratch] [Preview] 1,1 All
| -mypackage.Rche | y <- matrix(rnorm(50), 10, 5, dimnames=list(paste("g", 1:10, sep="
| |+00_pkg_src/ | # Row clustering
| |+mypackage/ | hr <- hclust(as.dist(1-cor(t(y), method="pearson")), method="compl
| |-00check.log | # Column clustering
| |-00install.ou | hc <- hclust(as.dist(1-cor(y, method="spearman")), method="complet
| |+mypackage-Ex | # Plot heatmap
| |+mypackage-Ex | heatmap.2(y, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), scale
| |+mypackage-Ex | # Return matrix with row/column sorting as in heatmap
| |+mypackage/ | y[rev(hr$labels[hr$order]), hc$labels[hc$order]]
| |+man/ | heat.colors
| |+colAg.Rd | heat.colors function grDevices
| |+mypackage- | heatmap function stats comple
| |+R/ | fms(sdfset[[1]], sdfset[[2]], fast=T)
| |+myfct.R | result <- fms(sdfset[[1]], sdfset[[2]])
| |-DESCRIPTION | fms <- fms(sdfset[[1]], sdfset[[2]], au=2, bu=1, matching.mode="a
| |-NAMESPACE | fms
| |+Read-and-del | script2.R [+] 12,1 33%
| |+fmsR_1.0.tar. |
| |+jitter.png |
| |+matrix.xls |
| |+myfct.R |
| |+mypackage_1.0. |
| |+notes.R |
| |+overLapper.R |
| |+rangeoverLappe |
| |+RNA-Seq.R |
| |+script1.R |
| |+script2.R |
| |+SDFstreamer.R |
| g9 0.1416941
| g10 0.5772262 0.3061073
| > as.matrix(c)[1:4,1:4]
| | g1 g2
| g1 1.0000000 -0.7240061
| g2 -0.7240061 1.0000000
| g3 0.8050921 -0.5586679
| g4 0.2327069 -0.7823333
| > y
| | t1 t2
| g1 -0.2608109 -2.1287458
| g2 -2.0478162 -0.2318061
| g3 -0.1814785 -0.5137189
| g4 0.2493454 -0.5782053
| g5 0.1082261 -1.8310231
| g6 0.2596634 -0.8048402
| g7 0.4497986 -0.6475571
| g8 -1.0501454 -0.3717143
| g9 -0.7831244 0.8490208
| g10 -1.7273262 0.3621398
| >
| >
| heatmap.2
| n
| Enhanced Heat Map
| Description:
| A heat map is a fals
| a
| dendrogram added to
| reordering of the ro
```

RStudio

Project: (None)

R demos *

Demos in package 'graphics':

- Hershey: Tables of the characters in the Hershey vector fonts
- Japanese: Tables of the Japanese characters in the Hershey vector fonts
- graphics: A show of some of R's graphics capabilities
- image: The image-like graphics builtins of R
- persp: Extended persp() examples
- plotmath: Examples of the use of mathematics annotation

Demos in package 'grDevices':

- colors: A show of R's predefined colors()
- hclColors: Exploration of hcl() space

Demos in package 'stats':

- glm.vr: Some glm() examples from V&R with several predictors
- lm.glm: Some linear and generalized linear modelling examples from 'An Introduction to Statistical Modelling' by Annette Dobson
- nlm: Nonlinear least-squares using nlm()
- smooth: 'Visualize' steps in Tukey's smoothers

Environment History

Global Environment

Variable	Value
g	Factor w/ 10 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1...
lev	num [1:12] 90 100 110 120 130 140 150 160 170 180 ...
n	100
opar	List of 1
pie.sales	Named num [1:6] 0.12 0.3 0.26 0.16 0.04 0.12
pin	num [1:2] 7.14 3.99
scale	0.00665185185185185
usr	num [1:4] 0.56 12.44 -0.082 2.132
x	num [1:87] 10 20 30 40 50 60 70 80 90 100 ...
xadd	106.567371937639

Files Plots Packages Help Viewer

Zoom Export Clear All

Edgar Anderson's Iris Data

```

> par(bg = "lightcyan")
> pin <- par("pin")
> xdelta <- diff(range(x))
> ydelta <- diff(range(y))
> xscale <- pin[1]/xdelta
> yscale <- pin[2]/ydelta
> scale <- min(xscale, yscale)
> xadd <- 0.5*(pin[1]/scale - xdelta)
> yadd <- 0.5*(pin[2]/scale - ydelta)
> plot(numeric(0), numeric(0),
+      xlim = range(x)+c(-1,1)*xadd, ylim = range(y)+c(-1,1)*yadd,
+      type = "n", ann = FALSE)
Hit <Return> to see next plot: |

```

RSTUDIO

Uno de los IDE's más utilizados

<https://www.rstudio.com>

```
hello.R *
Source on Save
Run Source
1 require(datasets)
2 require(grDevices)
3 require(graphics)
4
5 x <- stats::rnorm(50)
6
7 opar <- par(bg = "white")
8
9 plot(x, ann = FALSE, type = "n")
10
11 abline(h = 0, col = gray(.90))
12
13 lines(x, col = "green4", lty = "dotted")
14
15 points(x, bg = "limegreen", pch = 21)
16
17 title(main = "Un ejemplo",xlab = "Software Gurú Virtual Conference",col.main = "blue", col.lab = gray
18
19 par(bg = "gray")
20
11:31 (Top Level) R Script
```



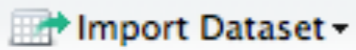


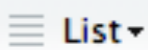
RSTUDIO

El código fuente (arriba-izquierda)

```
Console ~/ ↵
> require(grDevices)
> require(graphics)
> x <- stats::rnorm(50)
> opar <- par(bg = "white")
> plot(x, ann = FALSE, type = "n")
Hit <Return> to see next plot: abline(h = 0, col = gray(.90))
> lines(x, col = "green4", lty = "dotted")
> points(x, bg = "limegreen", pch = 21)
> title(main = "Simple Use of Color In a Plot",xlab = "Just a Whisper of a Label",col.main = "blue", col.l
ab = gray(.8),cex.main = 1.2, cex.lab = 1.0, font.main = 4, font.lab = 3)
> require(datasets)
> require(grDevices)
> require(graphics)
> x <- stats::rnorm(50)
> opar <- par(bg = "white")
> plot(x, ann = FALSE, type = "n")
Hit <Return> to see next plot: abline(h = 0, col = gray(.90))
> lines(x, col = "green4", lty = "dotted")
> points(x, bg = "limegreen", pch = 21)
> title(main = "Un ejemplo",xlab = "Software Gurú Virtual Conference",col.main = "blue", col.lab = gray(.8
),cex.main = 1.2, cex.lab = 1.0, font.main = 4, font.lab = 3)
> |
```

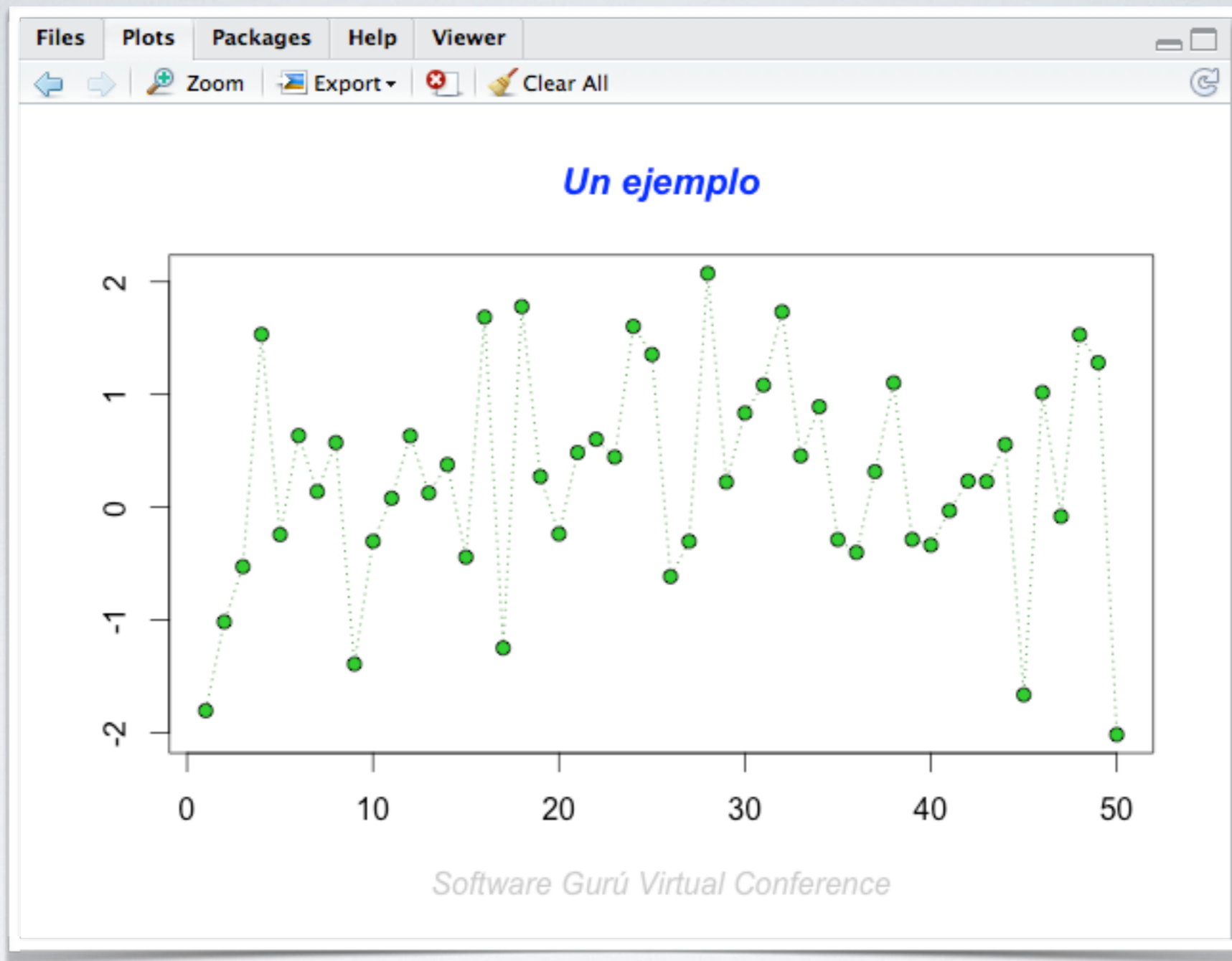
RSTUDIO

La consola, el shell de R (abajo-izquierda)

Environment		History	
    			
Global Environment ▾		<input type="text" value=""/>	
Values			
g	Factor w/ 10 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 1..		
lev	num [1:12] 90 100 110 120 130 140 150 160 170 180 ...		
n	100		
▶ opar	List of 1		
pie.sales	Named num [1:6] 0.12 0.3 0.26 0.16 0.04 0.12		
pin	num [1:2] 7.14 3.99		
scale	0.00665185185185185		
usr	num [1:4] -139 1019 -14 634		
x	num [1:50] -1.804 -1.017 -0.527 1.532 -0.243 ...		
xadd	106.567371937639		
xdelta	860		

RSTUDIO

Entorno (arriba-derecha)



RSTUDIO

Gráficas, paquetes, archivos, ayuda (abajo-derecha)



Live Collaborations

Live-R delivers live collaboration capabilities to enable geographically dispersed and multi-discipline colleagues and coworkers to seamlessly transition from individual productivity to higher-value team productivity.

[LEARN MORE](#)

Live-R - Enterprise-class, Cloud-based, 100% Web-delivered Data Analytics, Visualizations & Live Collaborations

[Learn More](#)

LIVE R

solución en la nube

<http://live-analytics.com>

Shiny in action

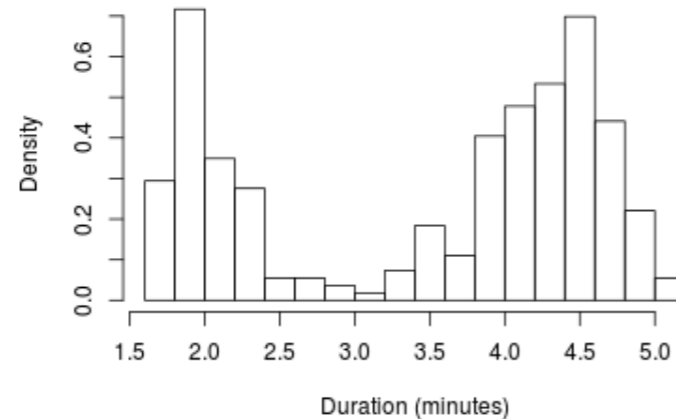
Here's a basic Shiny application, consisting of less than 40 lines of code.
Try changing the number of bins and toggling the checkboxes.

Number of bins in histogram (approximate):

20

- Show individual observations
- Show density estimate

Geyser eruption duration



ui.R [server.R](#)

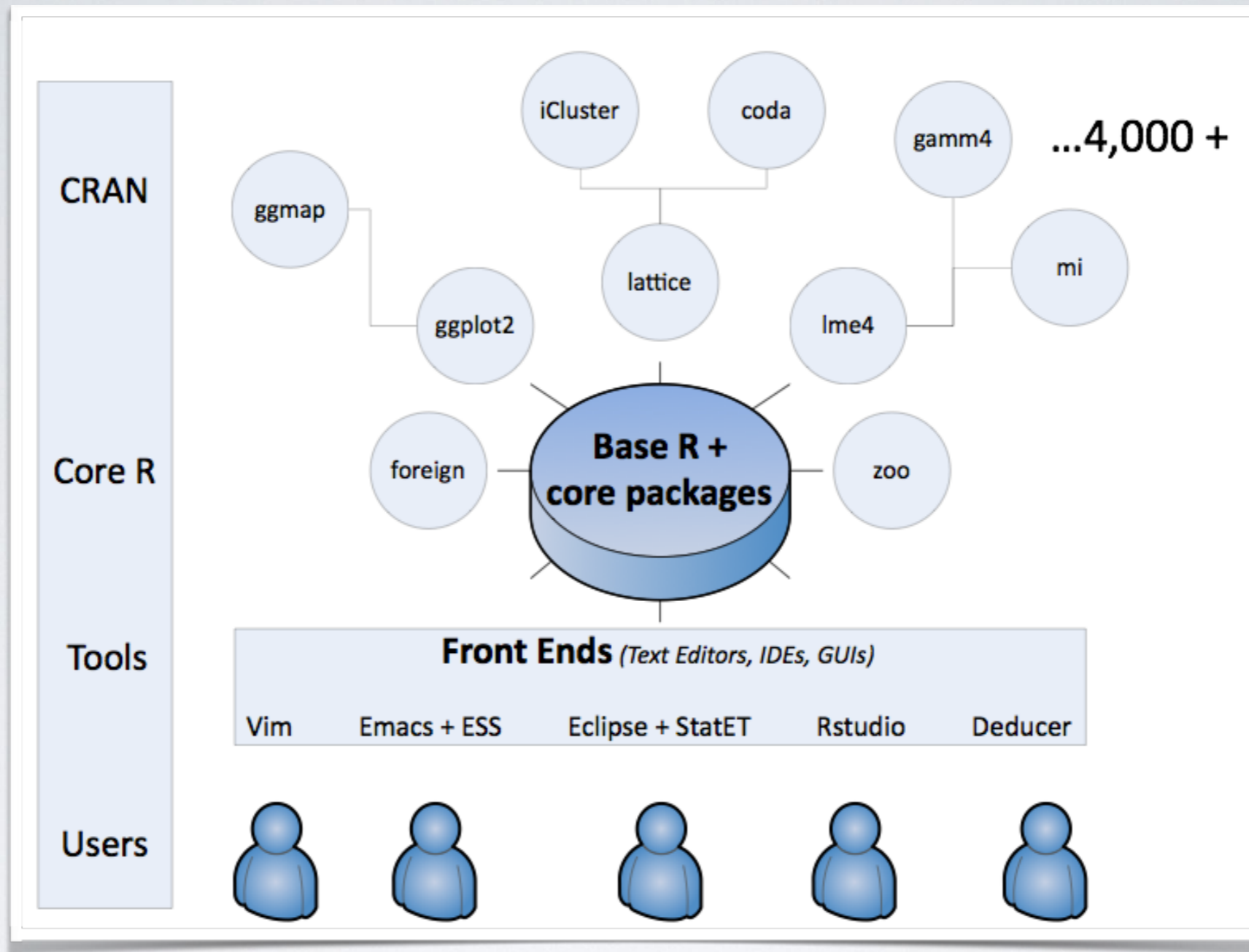
```
shinyUI(bootstrapPage(  
  selectInput(inputId = "n_breaks",  
    label = "Number of bins in histogram (approximate):",  
    choices = c(10, 20, 35, 50),  
    selected = 20),  
  checkboxInput(inputId = "individual_obs",  
    label = strong("Show individual observations"),  
    value = FALSE),  
  checkboxInput(inputId = "density",  
    label = strong("Show density estimate"),  
    value = FALSE),  
  plotOutput(outputId = "main_plot", height = "300px"),  
  # Display this only if the density is shown  
  conditionalPanel(condition = "input.density == true",  
    sliderInput(inputId = "bw_adjust",  
      label = "Bandwidth adjustment:",  
      min = 0.2, max = 2, value = 1, step = 0.2)  
  )  
)  
)
```

Please see our [tutorial](#) to learn more about writing Shiny apps.

SHINY

aplicaciones web con R

<http://www.rstudio.com/shiny>



MÁS QUE UN LENGUAJE

R ofrece más de 4,000 paquetes en <http://cran.r-project.org>

¿EL LENGUAJE?

```
> 1+2  
[1] 3
```

LENGUAJE INTERPRETADO

Usa un CLI, como Python y JavaScript

```
> x1      # vector
[1] 5
>
> x1      # escalar
[1] 5
>
> x2      # vector
[1] 1 2 3
>
> x3      # matriz
      [,1] [,2] [,3]
[1,]   1   2   3
[2,]  11  12  13
```

TODO ES UN VECTOR

(Todo)

MULTIPARADIGMA

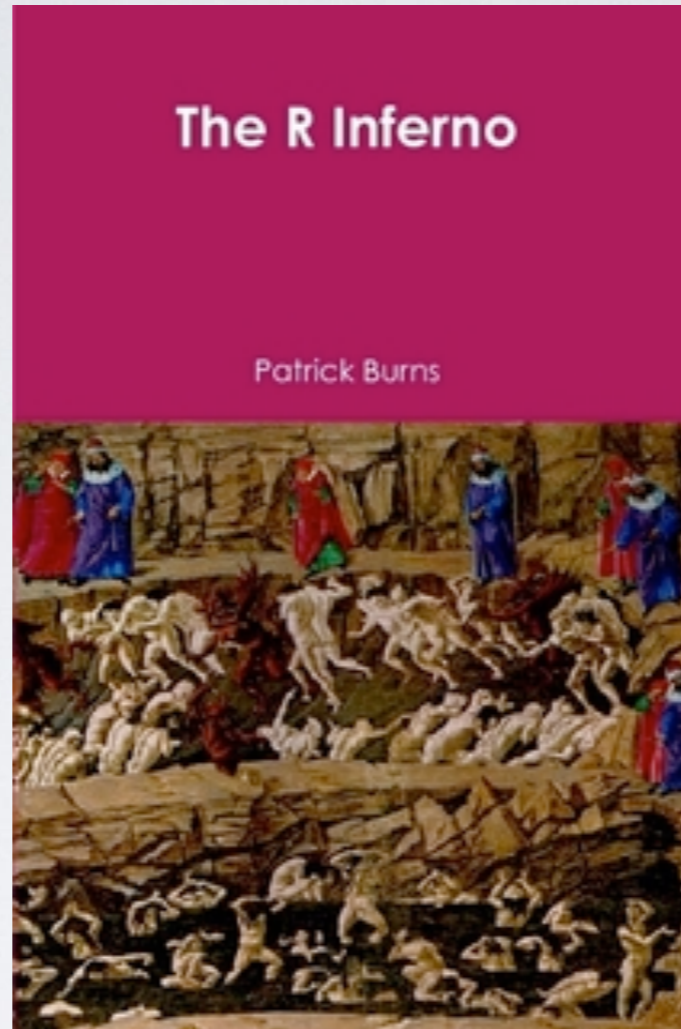
- **Imperativo** (como un script, ejecutando una instrucción tras otra, como Python)
- **Orientado a objetos** (estilo Java)
- **Funcional** (estilo Lisp, con funciones de *primera clase* — funciones que pueden ser pasadas como argumentos de otras funciones—)

MULTIPARADIGMA

- **Imperativo** (como un script, ejecutando una instrucción tras otra, como Python)
- **Orientado a objetos** (estilo Java)
- **Funcional** (estilo Lisp, con funciones de *primera clase* — funciones que pueden ser pasadas como argumentos de otras funciones—)

Como diría Larry Wall, esto sólo significa que...

“Hay más de una forma de hacerlo”



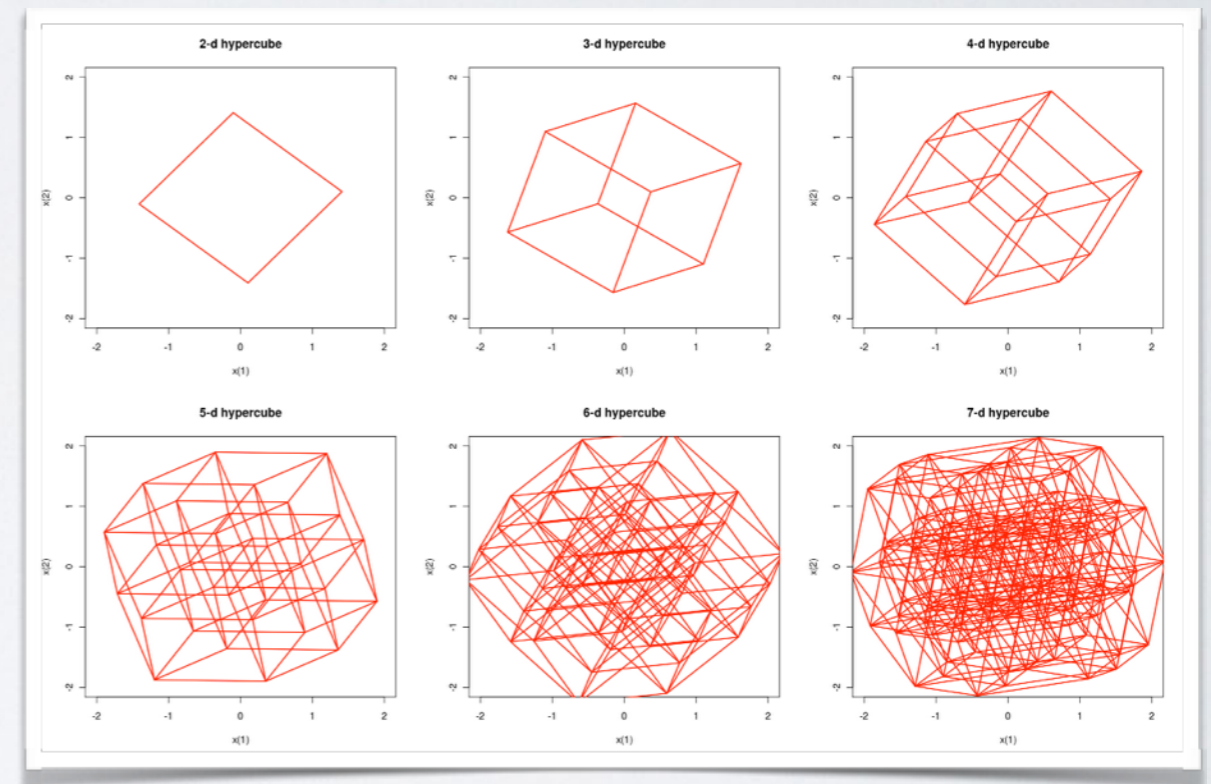
“EL PROBLEMA DE R ES
QUE FUE ESCRITO POR
GENTE DE ESTADÍSTICA”

MÁS QUE UN LENGUAJE

R es un entorno de programación para hacer estadística
(de ahí sus peculiaridades)

MÁS QUE UN LENGUAJE

- Más que un lenguaje, un entorno de interactivo de programación
- Pensemos en R más como un entorno que “**tiene** un lenguaje” y menos como “**es** un lenguaje”
- Así como Excel tiene VBA



R

y sus peculiaridades

ASIGNACIÓN

- La flecha \leftarrow es el operador de asignación

$$e \leftarrow m * c^2$$

- Aunque es posible esto, es raro...

$$m * c^2 \rightarrow e$$

GUIÓN BAJO

- Antes podías hacer esto para asignar el valor de 'f' a la variable 'e'

```
e_f
```

- Muuuy raro... Y esto llevó a usar el punto en el nombre de las variables

```
e.numero <- 2
```

- Eso desembocó en el uso de \$ tal como usamos el punto en otros lenguajes

```
e$columna
```

LETRAS “RESERVADAS”

- O que es mejor usar como tales

c, q, s, t, C, D, F, I, T

- Por ejemplo:
 - T y F son variables globales que valen por defecto TRUE y FALSE, respectivamente... Pero pueden cambiar :)
 - c es la función para crear vectores
 - q es la versión corta de *quit*, para terminar la sesión

VECTORES

- Es el tipo de dato primario en R
- Los vectores en R no son objetos matemáticos (elementos de un espacio vectorial)
- Los vectores en R son colecciones de datos, con operaciones uno-a-uno

```
> x <- c(1,2,3)
> y <- c(1,2,3)
> x*y
[1] 1 4 9
```

VECTORES

- Las operaciones entre vectores de diferentes longitudes son permitidas.

```
> y <- c(1,2,3,4)
```

```
> x+y
```

```
[1] 2 4 6 5
```

Warning message:

In x + y : longer object length is not a multiple of shorter object length

```
> y <- c(1,2,3,4,5,6)
```

```
> x+y
```

```
[1] 2 4 6 5 7 9
```

VECTORES

- No hay escalares en R, en todo caso, son vectores con un solo elemento.
- Los vectores tienen el mismo tipo de dato.

```
> c(1,2,3,4)
[1] 1 2 3 4
>
> c(1,2,3,4.1)
[1] 1.0 2.0 3.0 4.1
>
> c(1,2,3,"4")
[1] "1" "2" "3" "4"
```

VECTORES

- A diferencia de la mayoría de los lenguajes, el índice inicial es 1 y no 0 (como en FORTRAN)
- El índice 0 es *nada*, literalmente. Una constante lógica llamada NA en R que indica “valor perdido”. Es parecido al NULL, y None de otros lenguajes.
- (Pero R tiene su propio valor NULL: un objeto utilizado para indicar que hay valores indefinidos.)

```
> y
[1] 1 2 3 4
>
> y[0]
numeric(0)
> y[1]
[1] 1
> y[5]
[1] NA
```

VECTORES

- Los índices negativos también son extraños...
- En Python
 - $y = [1, 2, 3, 4]$
 - $y[-3]$ es 2
- En R
 - $y \leftarrow c(1, 2, 3, 4)$
 - $y[-3]$ es eliminar el tercer componente

VECTORES

- R tiene índices booleanos, y pueden ser muy útiles para seleccionar elementos

```
> x <- c(1,2,3,4,5,6,7,8)
> x[x>3]
[1] 4 5 6 7 8
```

- Y en asignaciones también

```
> x <- c(3, 1, 4, 1, 5, 9)
> x[x>3] <- 7
> x
[1] 3 1 7 1 7 7
```

TIPOS

- Un vector debe contener solo uno de estos tipos de datos atómicos
 - logical, integer, double, complex, character, raw.
- La función **as.** permite convertir tipos (*typecasting*)

```
> as.integer(2.3)
[1] 2
```

LISTAS

- Son como vectores, pero con diferentes tipos de datos
- Accedes a los elementos con `[[]]`

```
> x <- list(1,"a",2.3)
> x
[[1]]
[1] 1

[[2]]
[1] "a"

[[3]]
[1] 2.3
```


LISTAS

- Puedes nombrar elementos de la lista y llamarlos con \$

```
> a <- list(nombre="Juan", 4, foo=c(3,8,9))
> a
$nombre
[1] "Juan"

[[2]]
[1] 4

$foo
[1] 3 8 9

> a$nombre
[1] "Juan"
```

LISTAS

- Y agregar elementos nuevos

```
> a$nuevo <- "algo nuevo"  
> a  
$nombre  
[1] "Juan"  
  
[[2]]  
[1] 4  
  
$foo  
[1] 3 8 9  
  
$nuevo  
[1] "algo nuevo"
```

MATRICES

- Tipo de datos rectangular: columnas y filas, todas con el mismo tipo de datos básico
- O bien, un vector pero con el atributo *dim*: la dimensión expresada en filas por columnas

```
> matrix(1:15, 5)
      [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

```
> matrix(1:15, 3,5)
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15
```

DATA FRAMES

- Es la “hoja de cálculo” de R (salvo porque las operaciones aplican a toda una fila o columna)
- A diferencia de una matriz, mezcla nombres, números y todo tipo de datos
- Excelente para manipular CSV, XLS, etc.

DATA FRAMES

- Cómo construir un data frame a partir de vectores

```
> n <- c(1,2,3)
> s <- c("a", "b", "c")
> b <- c(TRUE, FALSE, TRUE)
> df <- data.frame(n,s,b)
> df
```

	n	s	b
1	1	a	TRUE
2	2	b	FALSE
3	3	c	TRUE

DATA FRAMES

- mtcars, un data frame muy completo que viene con R para hacer pruebas

```
> mtcars
```

```
      mpg  cyl  disp  hp  drat    wt  qsec vs
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46 0
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02 0
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61 1
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44 1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02 0
Valiant        18.1   6 225.0 105 2.76 3.460 20.22 1
Duster 360     14.3   8 360.0 245 3.21 3.570 15.84 0
Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00 1
```

DATA FRAMES

- Acceder a los datos

```
> mtcars[1,2]
[1] 6
> mtcars["Mazda RX4", "cyl"]
[1] 6
```

- Por filas o columnas

```
> mtcars["Mazda RX4",]
      mpg cyl disp  hp drat   wt  qsec vs am gear carb
Mazda RX4  21   6  160 110  3.9 2.62 16.46  0  1   4    4
> mtcars[, "cyl"]
[1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4
[29] 8 6 8 4
> mtcars$cyl
[1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4
[29] 8 6 8 4
```

UN RESUMEN

- Tres tipos de objetos:
 1. Vector
 2. Lista
 3. NULL
- Objetos derivados:
 1. Matriz
 2. Data Frame
 3. Factor (una forma de categorizar datos)

R

algo más

FUNCIÓNES

- Son como en JavaScript:
 - También son objetos, pero sin llamarles clases
 - Es POO basada en prototipos
- Puedes crear funciones anónimas (lambda)
- Los argumentos siempre son pasados por valor (salvo cuando son cambiados)


```
f <- function(a, b=10)
{
    return (a+b)
}
```

SECUENCIAS

```
> seq(0,10)
[1] 0 1 2 3 4 5 6 7 8 9 10
>
> 0:10
[1] 0 1 2 3 4 5 6 7 8 9 10
>
> seq(0,10,2)
[1] 0 2 4 6 8 10
>
> seq(0,10,2.2)
[1] 0.0 2.2 4.4 6.6 8.8
```

PAQUETES

- Los paquetes disponibles en su instalación con *library()*



```
R packages available (RlibraryQR402611b356b)
Help search

1 Packages in library '/Library/Frameworks/R.framework/Versions/3.0/Resources/library':
2
3 base      The R Base Package
4 boot     Bootstrap Functions
5          (originally by Angelo
6          Canty for S)
7 class    Functions for
8          Classification
9 cluster  Cluster Analysis Extended
10         Rousseeuw et al.
11 codetools Code Analysis Tools for R
12 colorspace Color Space Manipulation
13 compiler The R Compiler Package
14 datasets The R Datasets Package
15 foreign  Read Data Stored by
16         Minitab, S, SAS, SPSS,
17         Stata, Systat, Weka,
18         dBase, ...
19 graphics The R Graphics Package
20 grDevices The R Graphics Devices and
21         Support for Colours and
22         Fonts
23 grid     The Grid Graphics Package
24 KernSmooth Functions for kernel
25         smoothing for Wand & Jones
26         (1995)
27 lattice Lattice Graphics
28 manipulate Interactive Plots for
29         RStudio
30 MASS    Support Functions and
31         Datasets for Venables and
32         Ripley's MASS
33 Matrix  Sparse and Dense Matrix
34         Classes and Methods
35 methods Formal Methods and Classes
36 mgcv    Mixed GAM Computation
37         Vehicle with GCV/AIC/REML
38         smoothness estimation
39 nlme    Linear and Nonlinear Mixed
```

PAQUETES

- Paquetes del repositorio CRAN (¡más de 4,000!)
- Instalación con `install.packages("nombre del paquete")`

```
> install.packages("fortunes")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.itam.mx/bin/macosx/contrib/3.0/
fortunes_1.5-2.tgz'
Content type 'application/x-gzip' length 191017 bytes (186 Kb)
opened URL
=====
downloaded 186 Kb

The downloaded binary packages are in
  /var/folders/pm/3ztp1p297fn4p4vj8td23wcm0000gp/T//Rtmp88P1Q/
downloaded_packages
```

PAQUETES

- Para utilizar un paquete recién instalado en R con la función *require()*

```
> require(fortunes)
Loading required package: fortunes
> fortune()
```

As for elegance, R is refined, tasteful, and beautiful. When I grow up, I want to marry R.

-- Andy Bunn (in a discussion about whether R is simple, powerful and elegant)
R-help (May 2005)

MANTENIMIENTO

- Una nueva versión cada 6 meses
- *update.packages()* hace este trabajo

```
> update.packages()
cluster :
  Version 1.15.1 installed in /Library/Frameworks/R.framework/Versions/3.0/Resources/library
  Version 1.15.2 available at http://cran.rstudio.com
Update (y/N/c)? |
```

WORKSPACE

- Todo queda en RAM
- `ls()`, `rm()`
- `sessionInfo()` resume toda la sesión

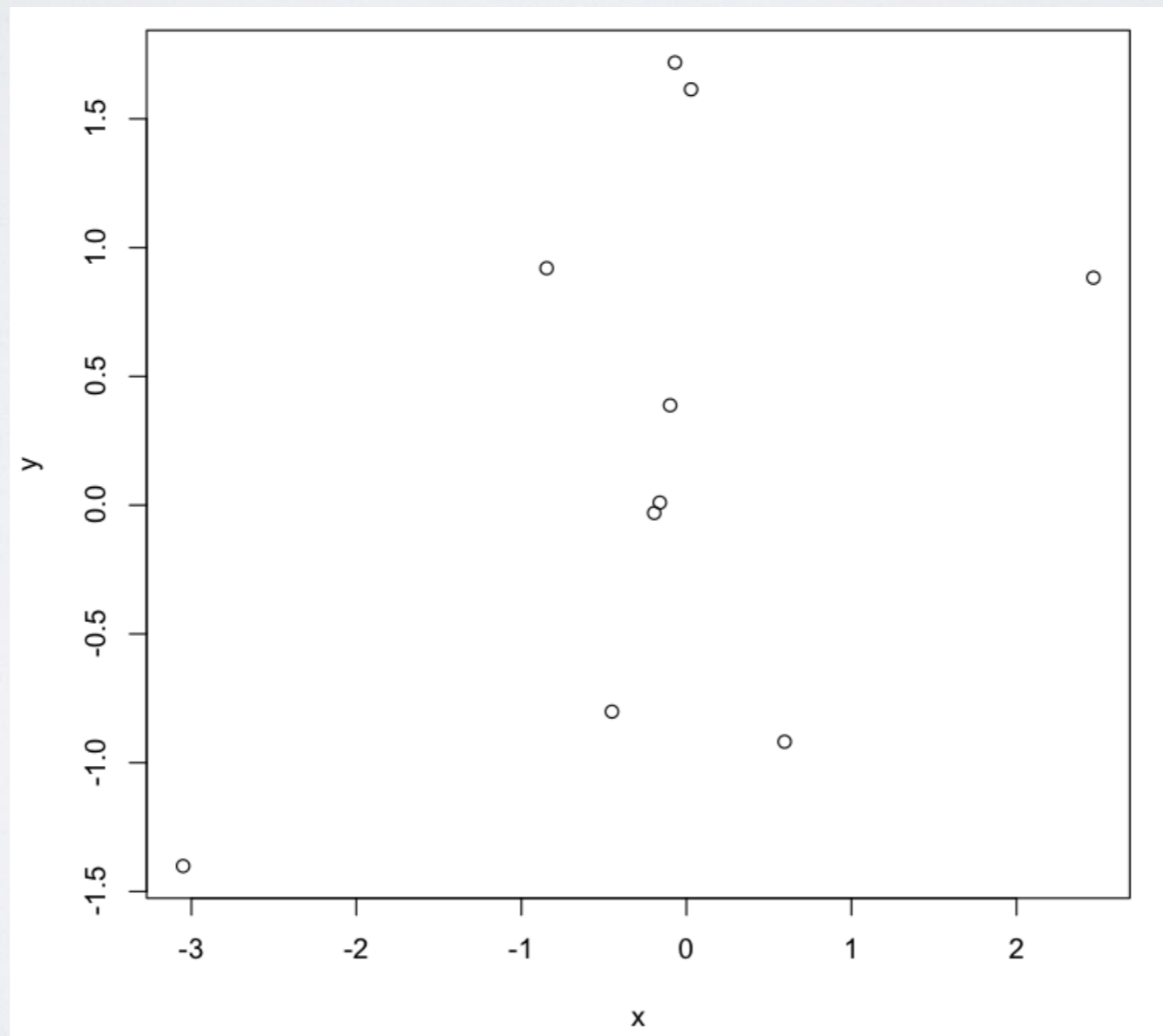
```
> x <- 5
> x
[1] 5
> z <- 3
> ls()
[1] "x" "z"
> ls.str()
x :  num 5
z :  num 3
> rm(x)
> ls()
[1] "z"
```


GRÁFICAS

- Nuestro lienzo es un *graphics device* creado principalmente con *plot()*
- Y sobre ese lienzo vamos pintando nuestra gráfica de forma muy personalizable
- Además de *plot()* tenemos *pie()*, *hist()*, *rect()*, *polygon()* y más funciones incluidas con R.
- R tiene varios paquetes para gráficas:
 - *rid*
 - *lattice*
 - *ggplot2* (muy popular)

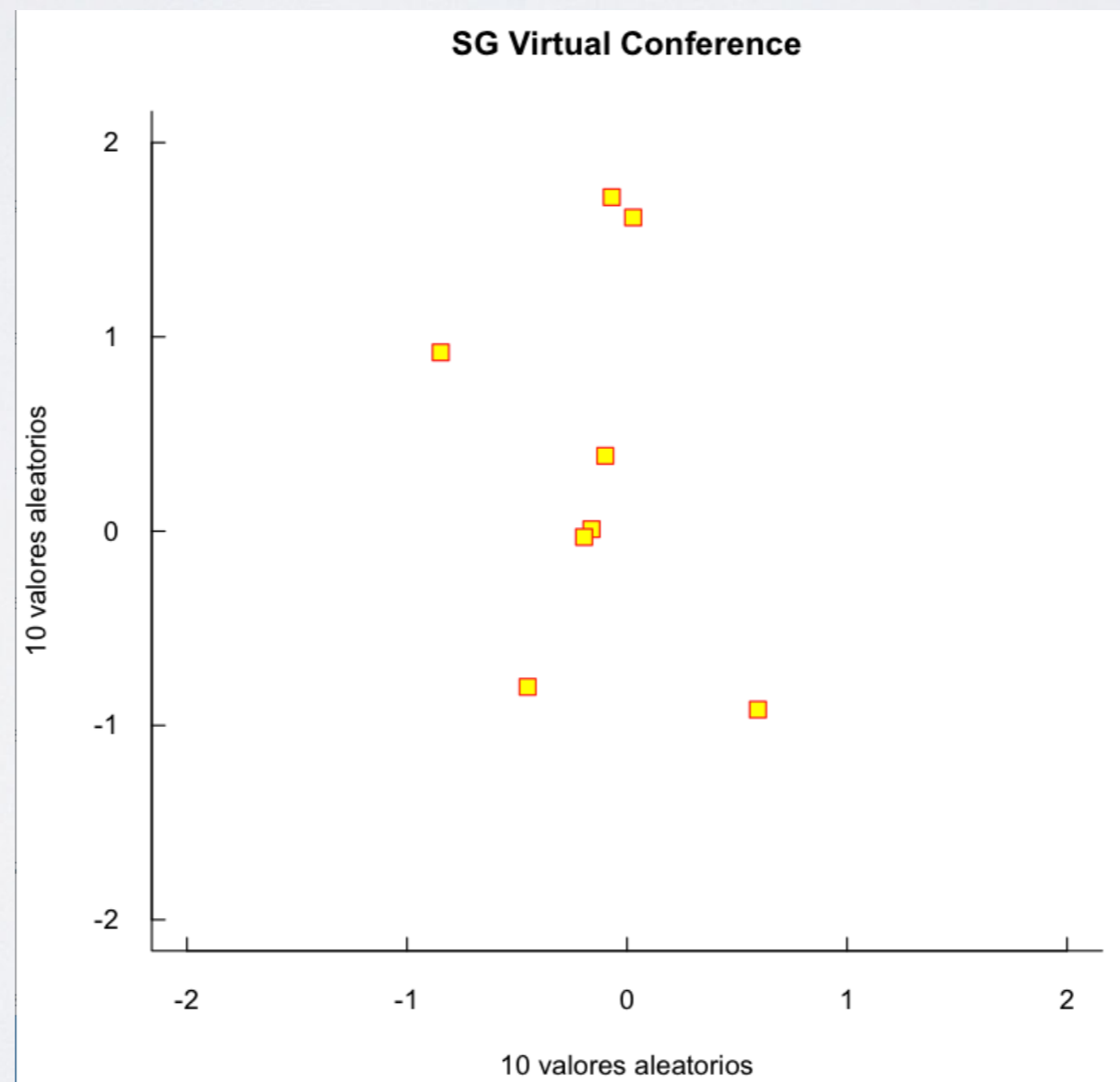
GRÁFICAS

```
> x <- rnorm(10)
> y <- rnorm(10)
> plot(x,y)
```



GRÁFICAS

```
> plot(x,y,xlab="10 valores aleatorios",  
ylab="10 valores  
aleatorios",xlim=c(-2,2),ylim=c(-2,2),pch=22  
,col="red",bg="yellow",bty="l",tcl=0.4,main=  
"SG Virtual Conference",las=1,cex=1.5)
```



ARCHIVOS

- **CSV**

- `data <- read.csv("datafile.csv")`

- **XLS**

- `library(gdata)`

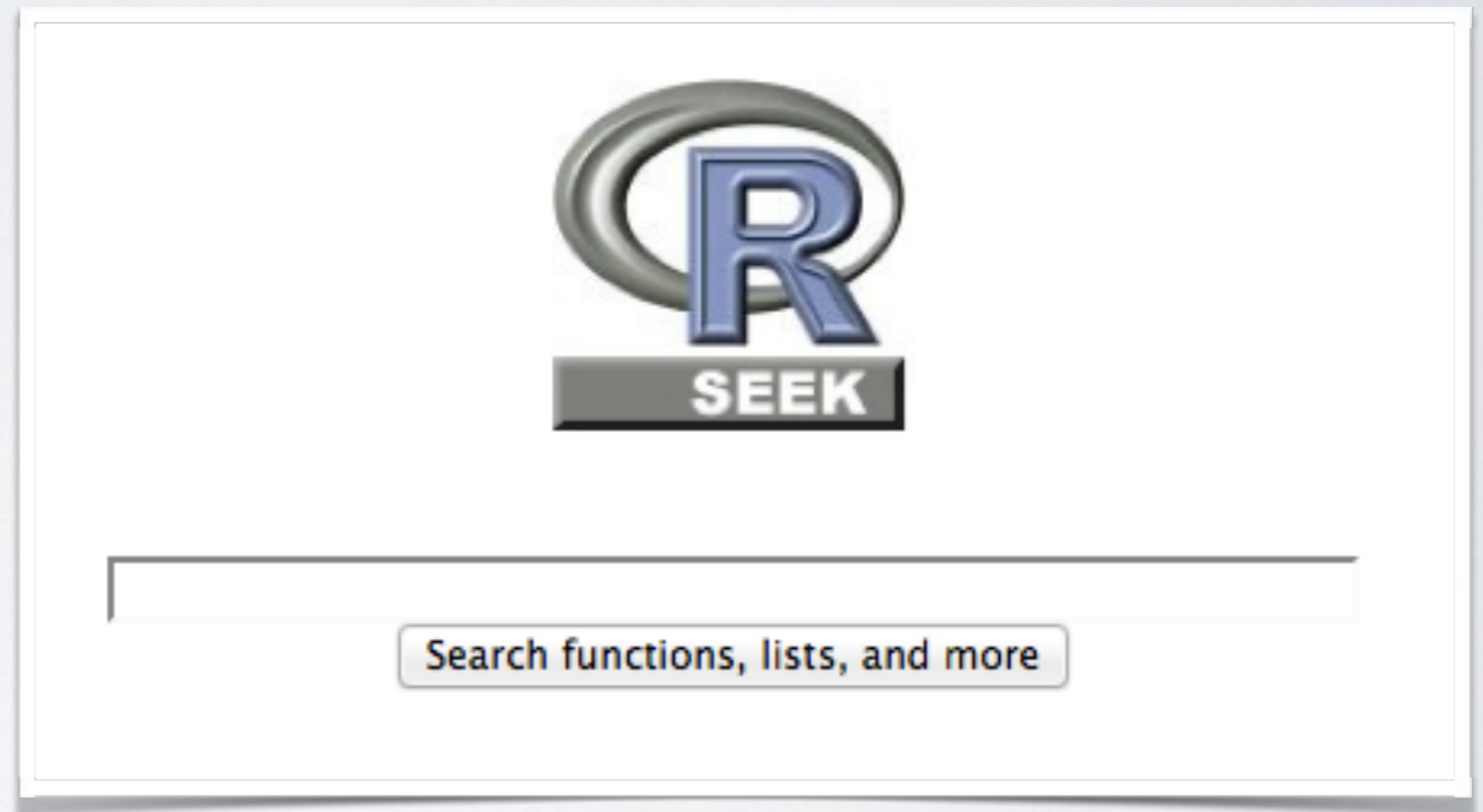
- `data <- read.xls("data.xls")`

- Donde la variable data es un data frame

- También abre archivos SPSS, TXT, XML, ...

AYUDA

- `help(plot)`
- `?plot`
- `example(plot)`
- <http://rseek.org>



CONCLUSIONES

- R es extraño (sintácticamente) para los programadores tradicionales
- R es software libre con una gran comunidad
- R es poderoso para hacer análisis de datos, con miles de paquetes disponibles

REFERENCIAS

- <http://data.princeton.edu/R/gettingStarted.html>
- [**http://www.revolutionanalytics.com**](http://www.revolutionanalytics.com)
- http://www.johndcook.com/R_language_for_programmers.html
- <http://www.ats.ucla.edu/stat/r/seminars/intro.htm>
- <http://jaredknowles.com/r-bootcamp/>
- [**http://tryr.codeschool.com/**](http://tryr.codeschool.com/)
- http://philipbjorge.com/dummy_wp_blog/wp-content/uploads/2012/03/RPoster.pdf
- <http://adv-r.had.co.nz/Vocabulary.html>
- <https://www.edx.org/course/mitx/mitx-15-071x-analytics-edge-1416>
- [**http://www.burns-stat.com/pages/Tutor/R_inferno.pdf**](http://www.burns-stat.com/pages/Tutor/R_inferno.pdf)
- [**http://cran.r-project.org/doc/contrib/Short-refcard.pdf**](http://cran.r-project.org/doc/contrib/Short-refcard.pdf)
- <http://www.r-tutor.com/>
- <http://baseballwithr.wordpress.com/about/>
- <http://www.burns-stat.com/documents/tutorials/impatient-r/>

SG 
VIRTUAL
CONFERENCE
6ta edición

Alan Lazalde



@alanlzd



alan.lazalde@gmail.com