



### ¿Qué es R?

R es un lenguaje y entorno para computación estadística y gráficos. Es un proyecto GNU que es similar al lenguaje y el entorno S que fue desarrollado en los Laboratorios Bell (anteriormente AT&T, ahora Lucent Technologies) por John Chambers y sus colegas. R puede considerarse como una implementación diferente de S. Hay algunas diferencias importantes, pero gran parte del código escrito para S se ejecuta sin alterar en R.

R proporciona una amplia variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, agrupación, ...) y técnicas gráficas, y es altamente extensible. El lenguaje S suele ser el vehículo elegido para la investigación en metodología estadística, y R proporciona una ruta de código abierto para participar en esa actividad.

Una de las fortalezas de R es la facilidad con la que se pueden producir gráficos de calidad de publicación bien diseñados, incluyendo símbolos matemáticos y fórmulas donde sea necesario. Se ha prestado gran atención a los valores predeterminados para las opciones de diseño menores en gráficos, pero el usuario mantiene el control total.

R está disponible como software libre bajo los términos de la Free Software Foundation's Licencia Pública General de GNU en forma de código fuente. Compila y se ejecuta en una amplia variedad de plataformas UNIX y sistemas similares (incluyendo FreeBSD y Linux), Windows y MacOS.



### ¿Qué es R?

- R-bloggers  
<https://www.r-bloggers.com/>
- Comunidad R  
<http://r-es.org/>
- Foro R eventualidades - preguntas  
<https://stackoverflow.com/questions/tagged/r>
- Manual Referencia  
<https://stat.ethz.ch/mailman/listinfo/r-help>
- R-DataMining  
<https://rdatamining.wordpress.com/>
- R-Documentation  
<https://www.rdocumentation.org/>



Robert Gentleman (izquierda) y Ross Ihaka (derecha) creadores de R.



#### El entorno R

R es un conjunto integrado de herramientas de software para la manipulación de datos, el cálculo y la visualización gráfica. Entre otras cosas tiene una instalación efectiva de manejo y almacenamiento de datos, un conjunto de operadores para cálculos sobre matrices, en particular matrices, una colección grande, coherente e integrada de herramientas intermedias para el análisis de datos, facilidades gráficas para el análisis y visualización de datos ya sea directamente en la computadora o en papel, y un lenguaje de programación bien desarrollado, simple y efectivo (llamado 'S') que incluye condicionales, bucles, funciones recursivas definidas por el usuario y facilidades de entrada y salida. (De hecho, la mayoría de las funciones proporcionadas por el sistema están escritas en el lenguaje S).

El término "entorno" pretende caracterizarlo como un sistema totalmente planificado y coherente, en lugar de una acumulación incremental de herramientas muy específicas e inflexibles, como suele ocurrir con otros programas de análisis de datos.

R es en gran medida un vehículo para los nuevos métodos de análisis interactivo de datos. Se ha desarrollado rápidamente y se ha extendido por una gran colección de paquetes . Sin embargo, la mayoría de los programas escritos en R son esencialmente efímeros, escritos para una sola pieza de análisis de datos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



#### Estadística con R

Hay alrededor de 25 paquetes suministrados con R (llamados paquetes "estándar" y "recomendados") y muchos más están disponibles a través de la familia de sitios de Internet CRAN (a través de <https://CRAN.R-project.org> ) y otros.

La mayoría de las estadísticas clásicas y gran parte de la metodología más reciente están disponibles para su uso con R, pero es posible que los usuarios deban estar preparados para hacer un poco de trabajo para encontrarlo.

Nota: Hay una diferencia importante en la filosofía entre S (y, por tanto, R) y los otros sistemas estadísticos principales. En S, un análisis estadístico se realiza normalmente como una serie de pasos, con resultados intermedios almacenados en objetos. De este modo, mientras que SAS y SPSS proporcionarán resultados abundantes de una regresión o análisis discriminante, R proporcionará un resultado mínimo y almacenará los resultados en un objeto adecuado para su posterior interrogación mediante otras funciones R.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Utilización Interactiva en R

Descargar el software

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

Comprehensive R Archive Network

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



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

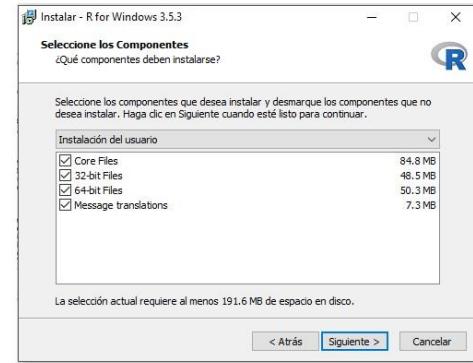
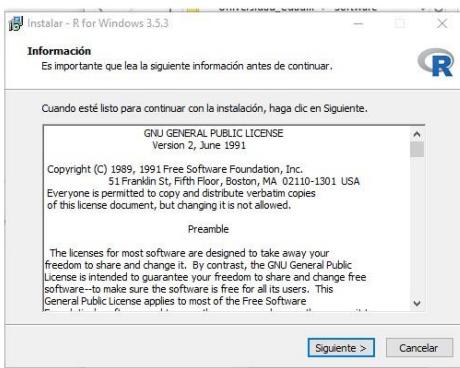
### Navigation

Current Issue  
Accepted articles  
Archive  
R News  
News and Notes  
Submissions  
Reviews and Proofreading  
Editorial Board

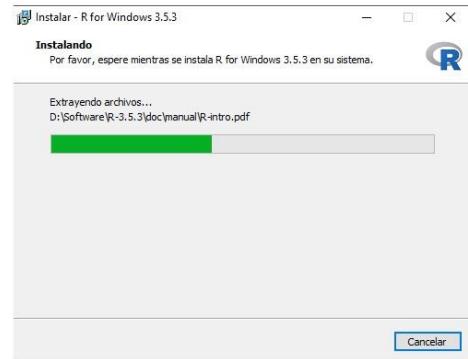
### Subscribe

RSS Feed   
ISSN: 2073-4859

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



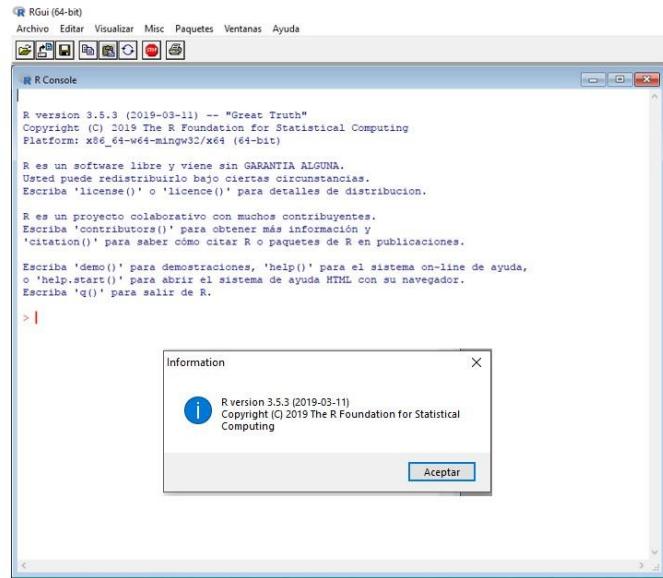
(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



```
pSWALDo dir
El volumen de la unidad D es DATOS
El número de serie del volumen es: B683-0DE2

Directorio de D:\Software\R-3.5.3\bin\x64

16/03/2019 08:02 p. m.    <DIR>                .
16/03/2019 08:02 p. m.    <DIR>                ..
11/03/2019 08:29 a. m.      18,944 open.exe
11/03/2019 08:29 a. m.      30,814,268 R.dll
11/03/2019 08:29 a. m.      104,960 R.exe
11/03/2019 08:29 a. m.      312,794 Rblas.dll
11/03/2019 08:29 a. m.      104,448 Rcmd.exe
11/03/2019 08:29 a. m.      88,576 Rfe.exe
11/03/2019 08:28 a. m.      322,525 Rgraphhapp.dll
11/03/2019 08:29 a. m.      88,576 Rgui.exe
11/03/2019 08:29 a. m.      66,785 Riconv.dll
11/03/2019 08:31 a. m.      2,741,760 Rlapack.dll
11/03/2019 08:29 a. m.      93,184 Rscript.exe
11/03/2019 08:29 a. m.      89,600 RSetReg.exe
11/03/2019 08:29 a. m.      88,576 Rterm.exe
13 archivos      34,934,936 bytes
2 dirs   194,512,580,600 bytes libres
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ordenes de R (Paquetes - Packages)

Todas las funciones R y conjuntos de datos se almacenan en paquetes . Sólo cuando se carga un paquete están disponibles sus contenidos. Esto se hace tanto para la eficiencia (la lista completa tomaría más memoria y tomaría más tiempo para buscar que un subconjunto), y para ayudar a los desarrolladores de paquetes.

Para ver qué paquetes están instalados, ejecute el comando  
> library ()

Para cargar un paquete en particular (por ejemplo, el paquete de boot que contiene funciones de Davison & Hinkley (1997))  
> library (boot)

Los usuarios conectados a Internet pueden usar las funciones install.packages() y update.packages()(disponibles a través del Packagesmenú en las GUI de Windows.

Para ver qué paquetes están cargados actualmente  
> search ()

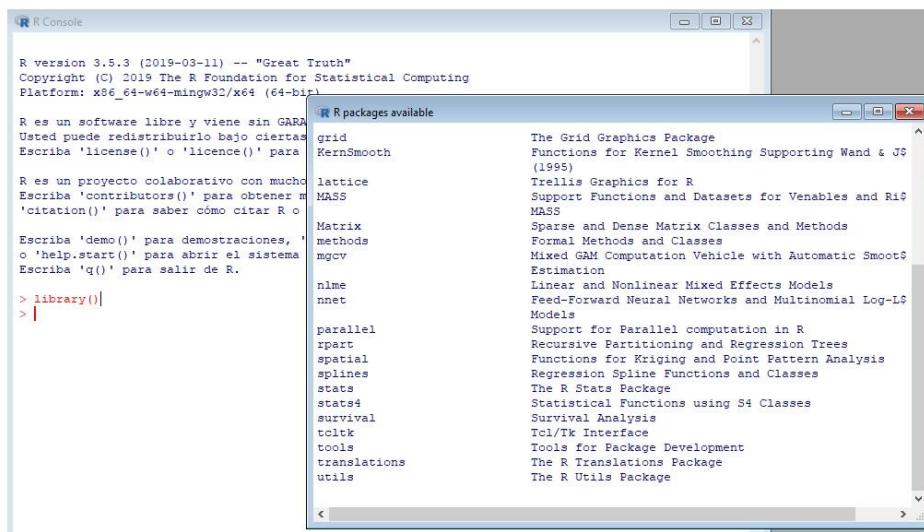
Para visualizar la lista de "search". Algunos paquetes pueden estar cargados pero no están disponibles en la lista.  
> loadedNamespaces ()

Para ver una lista de todos los temas de ayuda disponibles en un paquete instalado  
> help.start ()

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ordenes de R (Paquetes - Packages)



R version 3.5.3 (2019-03-11) -- "Great Truth"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (64-bit)

R es un software libre y viene sin GARANTÍAS.  
Usted puede redistribuirlo bajo ciertas condiciones.  
Escriba 'license()' o 'licence()' para obtener más información.

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

Escriba 'demo()' para demostraciones,  
o 'help.start()' para abrir el sistema de ayuda.  
Escriba 'q()' para salir de R.

```
> library()
> |
```

R packages available

grid	The Grid Graphics Package
KernSmooth	Functions for Kernel Smoothing Supporting Wand & JS (1995)
MASS	Trellis Graphics for R
Matrix	Support Functions and Datasets for Venables and Ripley's MASS
methods	Sparse and Dense Matrix Classes and Methods
mgcv	Formal Methods and Classes
nlme	Mixed GAM Computation Vehicle with Automatic Smoothing Estimation
nnet	Linear and Nonlinear Mixed Effects Models
parallel	Feed-Forward Neural Networks and Multinomial Log-Log Models
rpart	Support for Parallel computation in R
spatial	Recursive Partitioning and Regression Trees
splines	Functions for Kriging and Point Pattern Analysis
stats	Regression Spline Functions and Classes
stats4	The R Stats Package
survival	Statistical Functions using S4 Classes
tcltk	Survival Analysis
tools	Tcl/Tk Interface
translations	Tools for Package Development
utils	The R Translations Package
	The R Utils Package

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



boot: Bootstrap Functions (Originally by Angelo Canty for S)  
Functions and datasets for bootstrapping from the book "Bootstrap Methods and Their Application" by A. C. Davison and D. V. Hinkley (1997, CUP), originally written by Angelo Canty for S.

Version: 1.3-20  
Priority: recommended  
Depends: R (> 3.0.0), graphics, stats  
Suggests: MASS, survival  
Published: 2017-07-30  
Author: Angelo Canty [aut], Brian Ripley [aut, trl, cre] (author of parallel support)  
Maintainer: Brian Ripley <ripley at stats.ox.ac.uk>  
License: Unlimited  
NeedsCompilation: no  
Citation: boot citation info  
Materials: ChangeLog  
In views: Econometrics, Optimization, SocialSciences, Survival, TimeSeries  
CRAN checks: boot results  
Downloads: Reference manual:boot.pdf  
Package source: boot\_1.3-20.tar.gz  
Windows binaries: r-devel: boot\_1.3-20.zip, r-release: boot\_1.3-20.zip, r-oldrel: boot\_1.3-20.zip  
OS X binaries: r-release: boot\_1.3-20.tgz, r-oldrel: boot\_1.3-20.tgz  
Old sources: boot archive  
Reverse dependencies:  
Reverse depends: addhaz, ade3s, BACprior, bcROCsurface, bootES, bootfard, candidates, CircStats, classyfire, Comp2ROC, Compind, cramer, dabestr, dawai, DCluster, dhglm, equivalence, experiment, fishmethods, fishmove, frailtypack, gb, HPbayes, imputeYn, IUPS, mdhglm, MDSGUI, MIIRL, NegBinBetaBinreg, nFactors, Oncotree, ordinalCont, phaseIIMD, popKorn, prLogistic, PsntSim, QuantPsy, R2WinBUGS, random.polychor.pa, relogit, rtp, sampSurf, sunboot, textefext, timesboot, titan, tlm, truncSP, verification, wBoot  
Reverse imports: adegenet, apaTables, AICo, AssetCorr, bandit, biotools, bootLR, brainGraph, brlmr, BSagni, btergm, catenary, Causata, childss, chngpt, circular, circplex, ciTools, clickR, climexRemes, complimbro, CorrBin, Countr, crs, CsChange, dagitty, DAMisc, dening, DescTools, dmeq, discharge, disiever, distichocr, DLMttool, dsmooth, DSAlRM, dynamicchazard, DySeq, ed50, eggCounts, egoTERGM, emdi, energy, episensr, evoper, extremogram, finalfit, frai, gamRR, GeneralOaxaca, ggstatplot, gsg, hdcpa, hmi, ic.infer, iClick, InformativeCensoring, IPSUR, jndi, jocre, Kendall, jaeken, LCAextend, LEGIT, lga, lme4, MarginalMediation, mase, MBESS, mdpeer, medflex, mem, Metalinegrator, metapla, mev, microsyn, MIIVsem, MixedPsy, modifiednk, momentuHMM, moveHMM, MplusAutomation, mudfold, multiROC, multivwaycov, MVN, netdfuseR, np, OptimaRegion, OptM, pairwiseCI, paramtest, pastecs, pDR, perspectev, pgirmess, phylosignal, phylometrics, plsr, plsRbeta, plsRglm, PointFore, polyapost, PopED, popper, pse, psychmetr, pubmed.mineR, qat, Qtools, R2OpenBUGS, rlineups, randomLCA, rapt, rcompanion, referenceIntervals, refund, restruktur, rich, robmed, robitxglm, RV.AideMemoire, SCBmeanfd, sem, sensitivity, sensobol, Sequential shotGroups, summar, simpleboot, sisal, SiZer, soiphysics, SongEvo, sparkTable, SpatialVx, spdep, SPRefUGED, strandCet, tbfr, TraMineR, treeclim, tsBSS, yhat, yuma, ZOI  
Reverse suggests: abd, AER, aod, asbio, bayesboot, broom, CaDENCE, car, CARBayes, ChemometricsWithR, cond, crossReg, DAAG, dominanceanalysis, doRedis, EffectsRelBaseline, ElemStatLearn, emplik, EnvStats, GEVcdn, glmmTMB, gMCP, Greg, GSIF, gaubfh, harvest, hoa, hrIPW, HSAUR, HSAUR2, investi, ipw, ISERtools, jaatha, jtools, LambertW, logmult, marg, matrixpls, meboot, metafor, mirt, mistat, nlreg, npsm, optimatch, proftools, ptest, R2HTML, Remdr, RcmdrMisc, riskRegression, rms, robustbase, scdhlm, semPLS, semTools, table1, Wate, WeightIt  
Linking:  
Please use the canonical form <https://CRAN.R-project.org/package=boot> to link to this page.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ordenes de R (Paquetes - Packages)

The screenshot shows the R graphical user interface. On the left, there's a sidebar with the following menu items: Paquetes, Ventanas, Ayuda. Under Paquetes, there are several options: Cargar paquete..., Seleccionar espejo CRAN..., Seleccionar repositorios..., Instalar paquetes(s)..., Actualizar paquetes..., and Install package(s) from local files... The 'boot' package is highlighted in blue in a dropdown menu titled 'Select one' on the right. The main window is titled 'R Console' and displays the R version 3.5.3 startup message. Below the console, there's a command-line input area with some R code and a status bar at the bottom indicating 'Version 3.5.3 (2019-03-11)'.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Clasificación de Paquetes

Clasificación de paquetes en CRAN	
Bayesian	Estadística bayesiana
ChemPhys	Quimiometría y Física Computacional
ClinicalTrials	Diseño, monitorización y análisis de ensayos clínicos
Cluster	Análisis de conglomerados y modelos mixtos
Distributions	Distribuciones
Econometrics	Econometría
Environmetrics	Modelos para la Ecología
ExperimentalDesign	Diseño de experimentos
Finance	Análisis financiero
Genetics	Modelos para la genética
Graphics	Gráficos
gR	Modelos de gRafo
High Performance Computing	Computación intensiva
MachineLearning	Aprendizaje maquina
MedicalImaging	Ánálisis de imágenes médicas
Multivariate	Ánálisis multivariado
NaturalLanguageProcessing	Procesamiento lenguaje natural
Optimization	Estadística operativa
Pharmacokinetics	Datos farmaquíméticos
Psychometrics	Modelos psicométricos
Robust	Métodos robustos
SocialSciences	Estadística aplicada a las CC.SS.
Spatial	Ánálisis de datos espaciales
Survival	Ánálisis de supervivencia
TimeSeries	Ánálisis de series temporales

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Namespaces

Los paquetes tienen Namespaces, que hacen tres cosas: permiten que el escritor de paquetes oculte funciones y datos que están destinados solo para uso interno, evitan que las funciones se rompan cuando un usuario (u otro escritor de paquetes) elige un nombre repetido con uno en el paquete, y proporcionan una manera de referirse a un objeto dentro de un paquete en particular.

Por ejemplo, t() es la función de transposición en R, pero los usuarios pueden definir su propia función con el nombre t. Los espacios de nombres impiden que la definición del usuario tenga prioridad y rompa todas las funciones que intentan transponer una matriz.

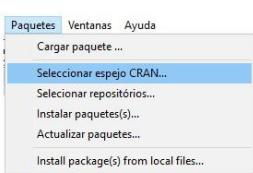
Hay dos operadores que trabajan con espacios de nombres. El operador de dos puntos dobles :: selecciona definiciones de un espacio de nombres particular. En el ejemplo anterior, la función de transposición siempre estará disponible como base::t, porque está definida en el base paquete. Solo las funciones que se exportan del paquete se pueden recuperar de esta manera.

El operador de tres puntos ::: se puede ver en algunos lugares en el código R: actúa como el operador de dos puntos, pero también permite el acceso a objetos ocultos. Es más probable que los usuarios utilicen la getAnywhere() función, que busca varios paquetes.

Los paquetes suelen ser interdependientes, y cargar uno puede hacer que otros se carguen automáticamente. Los operadores de dos puntos descritos anteriormente también causarán la carga automática del paquete asociado. Cuando los paquetes con espacios de nombres se cargan automáticamente, no se agregan a la lista de búsqueda

```
R Console
> utils:::menuInstallLocal()
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



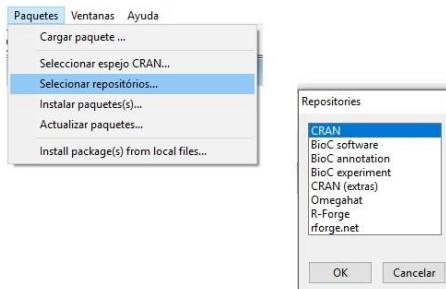
#### Comprehensive R Archive Network

CRAN es una red de servidores ftp y web de todo el mundo que almacena versiones idénticas y actualizadas de código y documentación para R.



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

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



R-Forge ofrece una plataforma central para el desarrollo de paquetes R, software relacionado con R y otros proyectos. Se basa en que FusionForge en SVN, paquetes comprobados y creados diariamente, listas de correo, seguimiento de errores, foros / foros, alojamiento de sitios, archivo permanente de archivos, copias de seguridad completas y administración total basada en la web.

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

#### Journal

INVITED SECTION: THE FUTURE OF R

9

#### Collaborative Software Development Using R-Forge

#### Software Map

Tag cloud | Project Tree | Project List

ANOVA Bayesian Bioinformatics Biostatistics Classification Clustering Copay number DNA Ecology Economics Finance GUI Genetic Algorithms Genetics HTML Machine Learning Machine learning Mixture Multiple Comparisons Multivariate Analysis Multivariate Regression Multivariate Techniques Next generation Sequencing ODE Phylogeny R R-Forge Rcmdr Regression SNP Test Theory Time series analysis Trajectory visualization distributions diversity dynamic systems ecology econometrics economics epidemiology finance gene expression generalized linear models genetics graphics high dimensional data likelihood linear models linear programming machine learning microarray missing data missing values mixed effect models mixed model model comparison model estimation model selection movement multivariate multivariate regression multivariate statistics non-parametric nonlinear models nonparametric optimization parametric model permutation tests phylogeny plotting prediction psychology raster regression reporting robust robust statistics simulation soil spatial spatial autocorrelation spatial classes spatial data spatial methods spatial point patterns spatial regression spatio-temporal species distribution models stochastic processes survival teaching text mining time series visualization

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



RForge se esfuerza por proporcionar un entorno colaborativo para los desarrolladores de paquetes R. El objetivo final es ofrecer servicios similares a SourceForge (como el repositorio SVN, el lugar para la documentación, las descargas, las listas de correo, bugzilla, wiki, etc.) sin el aspecto molesto pero con características adicionales específicas para el desarrollo del paquete R, como make verifica la compilación, casi todas las compilaciones de paquetes, las pruebas en varias plataformas y el acceso completo a la repositorio similar a CRAN. El enfoque está en las características específicas de R que no son ofrecidas por SourceForge o GForge. Es complementario a sitios como GitHub con los que se puede integrar como back-end de paquetes R.

**Current projects**

Name	Title
Acinonyx	Plot & Animate: web-generating interactive graphics
adogram	Single and double-headed conditional adograms
affine	Functions and datasets for Web-based related knowledge
ALAAR	Atlas of Living Australia (ALA) data and resources in R.
animation	The animation package
AntBioR	
Arduino	Arduino projects
audio	Audio interface for R
background	Background processing and asynchronous callbacks
base64enc	Tools for base64 encoding
brew	Managing and R code output for report generation
CADStat	Provides a GUI to several statistical methods for making biological inferences
Cairo	Graphics device using Cairographics library for creating bitmap images or output
carbon	Carbon event loops for R
CarbonEL	Benchmark functions for the CEC 2013 Special Session on Real-Parameter Optimization
cec2013	Configuration Function Analysis with Design Matrix
cfc2	Functions for communicating climate, weather and extreme event information
climdesc	Consensus clustering methodology using re-sampling statistics
clusterCens	Primitive Obj-C/Cocoa frontend
Cocoa	Conservation planning optimization
ConsPlan	

## Arduino

About Arduino  
SVN access  
Download/Files

## Files

This project has no custom files.

## Most Recent Snapshots

[Arduino\\_rev4.tar.gz](#) 18,379 bytes 2014-03-29 13:39



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



**Paquetes** Ventanas Ayuda

Cargar paquete ...

Seleccionar espejo CRAN...

**Seleccionar repositorios...**

Instalar paquetes(s)...

Actualizar paquete(s)...

Install package(s) from local files...

**Repositories**

- CRAN**
  - BioC software
  - BioC annotation
  - BioC experiment
  - CRAN (extras)
  - Omegahat
  - R-Forge
  - rforge.net

OK Cancelar

The Omega Project for Statistical Computing  
<http://www.omegahat.net/>  
<http://www.omegahat.net/R/src/contrib/>

### Ejemplo

RFirefox: Esto incrusta R dentro del navegador web Firefox y permite que el código JavaScript llame a las funciones R y evalúe las expresiones R. También permite que las funciones y el código de R tengan acceso a los objetos y métodos de JavaScript y, por lo tanto, se utilicen como controladores de eventos y generen contenido dinámico dentro de una página web. (Gabe Becker y Duncan Temple Lang)



<https://www.bioconductor.org/install/>

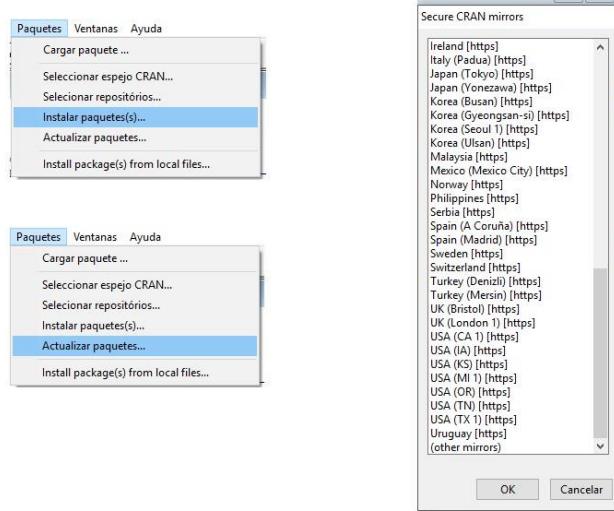
<https://www.youtube.com/user/bioconductor>

## Docker containers for Bioconductor

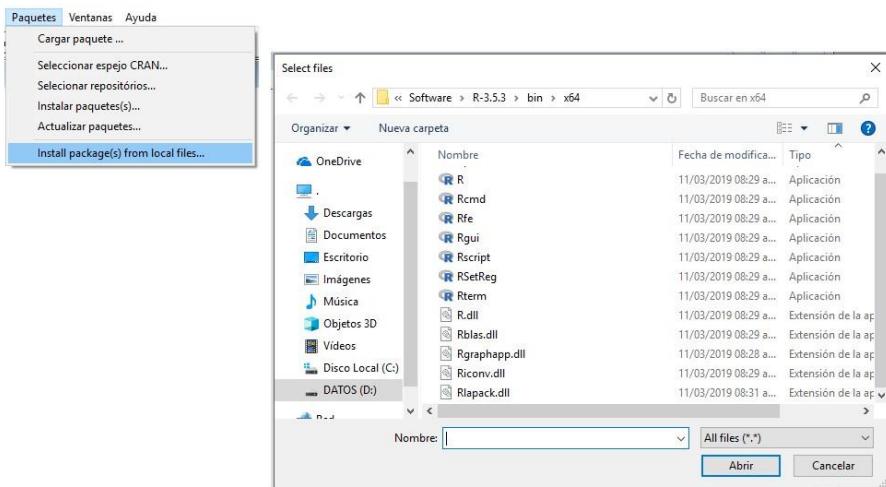
Docker allows software to be packaged into containers: self-contained environments that contain everything needed to run the software. Containers can be run anywhere (containers run in modern Linux kernels, but can be run on Windows and Mac as well using a virtual machine called Docker Toolbox). Containers can also be deployed in the cloud using Amazon EC2 Container Service or other cloud providers.

- [Why Use Containers](#)
- [Current Containers](#)
- [Legacy Containers](#)
- [Using Containers](#)
  - [Running Containers](#)
  - [Mounting Additional Volume](#)
- [Modifying Image Container](#)
- [Default package with Core2 Container](#)
- [Acknowledgements](#)

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Comandos de Consola

Desplazarse.  
Teclado: PgUp, PgDown, Ctrl+Arrows, Ctrl+Home, Ctrl+End, Mouse: use la barra de desplazamiento.

Edición.  
Flecha izquierda o Ctrl+B: mueve un carácter para atrás; Flecha derecha o Ctrl+F: mueve un carácter para adelante; Home o Ctrl+A: va al principio de línea; End o Ctrl+E: va al fin de línea; Histórico: flechas arriba y abajo, Ctrl+P, Ctrl+N  
Borrado.  
Del o Ctrl+D: borra carácter actual o sección; Backspace: delete preceding character; Ctrl+Delete o Ctrl+K: borra texto desde carácter actual al fin de línea.  
Ctrl+U: borra todo el texto desde la línea actual.  
Copiar y pegar.  
Usa el mouse (con el botón izquierdo sujetado) para marcar (seleccionar) texto.  
Use Shift+Del (o Ctrl+C) para copiar el texto seleccionado para área de transferencia.  
Shift+Ins (o Ctrl+V) para pegar el contenido del área de transferencia (si hubiera alguno)  
para la consola, Ctrl+X: primera copie después pegue  
Métodos.  
Ctrl+L: Limpia la consola.  
Ctrl+O o INS: Alterna o modo de sobre escritura:  
inicialmente desactivado.  
Ctrl+T: Intercambia carácter actual con uno de la izquierda.  
Nota: Consola es actualizada solo cuando alguna entrada es requerida.  
Use Ctrl+W para alternar esta característica  
prendido/apagado.  
Use ESC para parar el interpretar.  
TAB comienza a completar la palabra.  
Teclas estandares de Windows pueden ser utilizadas para cambiar a  
dispositivo gráfico (Ctrl+Tab ou Ctrl+F6 en MDI, Alt+Tab en  
SDII)

```
R Gui (64-bit)
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
R Console
> help
function (topic, package = NULL, lib.loc = NULL, verbose =getOption("verbose"),
try.all.packages =getOption("help.try.all.packages"), help_type =getOption("help_type"))
{
  types <- c("text", "html", "pdf")
  help_type <- if (!length(help_type))
    "text"
  else match.arg(tolower(help_type), types)
  if (!is.missing(package))
    if (is.name(y <- substitute(package)))
      package <- as.character(y)
  if (missing(topic)) {
    if (!is.null(package)) {
      if (interactive() && help_type == "html") {
        port <- tools:::startDynamicHelp(NA)
        if (port <= 0L)
          return(library(help = package, lib.loc = lib.loc,
                        character.only = TRUE))
        browser <- if (.Platform$GUI == "AQUA") {
          get("aqua.browser", envir = as.environment("tools:RGUI"))
        }
        elsegetOption("browser")
        browseURL(paste0("http://127.0.0.1:", port, "/library/",
                         package, "/html/00Index.html"), browser)
        return(invisible())
      }
      else return(library(help = package, lib.loc = lib.loc,
                          character.only = TRUE))
    }
    if (!is.null(lib.loc))
      return(library(lib.loc = lib.loc))
  }
  topic <- "help"
  package <- "utils"
  lib.loc <- .Library
}
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



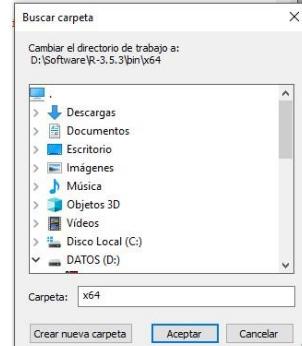
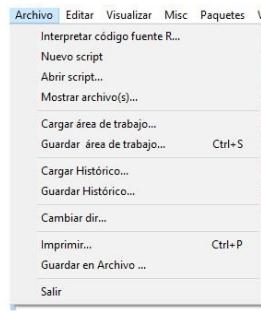
## Ejecución de ordenes desde un archivo

R tiene facilidades bastante extensas para acceder al sistema operativo bajo el cual se está ejecutando: esto permite que se use como un lenguaje de scripting y esa capacidad es muy utilizada por R, por ejemplo, para instalar paquetes.

Debido a que los scripts de R deben funcionar en todas las plataformas

### Ejemplo

```
> local({fn<-choose.files(filters=Filters[c('R','txt','All')],index=4)
+ file.show(fn,header=fn,title="")})
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Archivos y directorios

Hay muchas funciones para manipular archivos y directorios. Aquí hay algunos punteros a algunos de los más utilizados.

Para crear un archivo o directorio (vacío), use `file.create` o `dir.create`. (Estos son los análogos de las utilidades POSIX `touch` y `mkdir`.) Para los archivos y directorios temporales en el directorio de la sesión R, consulte `tempfile`.

Los archivos pueden eliminarse mediante : `file.remove` o `unlink`; este último puede eliminar los árboles de directorios.

Para listados de directorios use `list.files` (también disponible como `dir`) o `list.dirs`. Estos pueden seleccionar archivos usando una expresión regular: para seleccionarlos mediante el uso de comodines `Sys.glob`.

Muchos tipos de información en una ruta de archivo (incluyendo, por ejemplo, si es un archivo o directorio) se pueden encontrar en `file.info`.

Hay varias formas de averiguar si un archivo 'existe' (un archivo puede existir en el sistema de archivos y no ser visible para el usuario actual). Existen funciones `file.exists`, `file.access` y `file_test` con varias versiones de esta prueba: `file.tests` una versión del POSIX `test` comando para aquellos familiarizados con los scripts de shell.

La función `file.copy` es el análogo R del comando POSIX `cp`.

La elección de archivos se puede hacer de forma interactiva `file.choose`: el puerto de Windows tiene las funciones más versátiles `choose.files` y `choose.dir`; hay funciones similares en el paquete `tcltk`: `tk_choose.files` y `tk_choose.dir`.

Funciones `file.show` y `file.edit` mostrarán y editarán uno o más archivos de una manera apropiada para el puerto R, utilizando las facilidades de una consola (como `RGui` en Windows o `R.app` en macOS) si está en uso.

Hay algún soporte para enlaces en el sistema de archivos: ver funciones `file.link` y `Sys.readlink`.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Filepaths

Con algunas excepciones, R confía en las funciones subyacentes del sistema operativo para manipular las rutas de archivo. Algunos aspectos de esto pueden depender del sistema operativo, y lo hacen, incluso hasta la versión del sistema operativo. Existen estándares de POSIX sobre cómo los sistemas operativos deben interpretar las rutas de acceso a archivos y muchos usuarios de R asumen el cumplimiento de POSIX; pero Windows no pretende ser compatible y otros sistemas operativos pueden ser menos que completamente compatibles.

Los siguientes son algunos problemas que se han encontrado con `filepaths`.

Los sistemas de archivos POSIX distinguen entre mayúsculas y minúsculas, por lo que `foo.png` y `Foo.PNG` son archivos diferentes. Sin embargo, los valores predeterminados en Windows y macOS no deben distinguir entre mayúsculas y minúsculas, y los sistemas de archivos FAT (comúnmente utilizados en el almacenamiento extraible) normalmente no distinguen entre mayúsculas y minúsculas (y todas las rutas de archivo pueden asignarse a minúsculas).

Casi todos los servicios de sistema operativo de Windows admiten el uso de barra o barra invertida como separador de ruta de archivo, y R convierte las excepciones conocidas al formulario requerido por Windows.

El comportamiento de las rutas de archivo con una barra inclinada depende del sistema operativo. Tales rutas no son válidas en Windows y no se debe esperar que funcionen. POSIX-2008 requiere que dichas rutas coincidan solo con los directorios, pero las versiones anteriores también les permitían coincidir con los archivos. Así que es mejor evitarlos.

Múltiples barras en `filepaths` como `/a/B/C/D` e `F` son válidos en los sistemas de archivos POSIX y se tratan como si solo hubiera una barra diagonal. Ellos son generalmente aceptados por funciones del sistema operativo de Windows. Sin embargo, las barras diagonales iniciales pueden tener un significado diferente.

Filepaths UNC de Windows (como `\server\dir1\dir2\file` y `?\UNC\server\dir1\dir2\file`) no son compatibles, pero pueden funcionar en algunas funciones R. Los sistemas de archivos POSIX pueden tratar una barra diagonal doble especialmente.

Windows permite rutas de archivo que contienen unidades y relativas al directorio actual en una unidad, por ejemplo, `d:/foo/bar` se refiere a `d:/a/b/c/foo/bar` si el directorio actual en la unidad `re:` es `/a/B/C`. Se pretende que funcionen, pero el uso de rutas absolutas es más seguro.

Funciones `basename` y `dirname` parten seleccionadas de una ruta de archivo: la forma recomendada para ensamblar una ruta de archivo desde componentes es `file.path`. La función `path.expand` hace 'expansión tilde', sustituyendo los valores por directorios de inicio (el usuario actual, y quizás los de otros usuarios).

En los sistemas de archivos con enlaces, muchas rutas de archivo pueden hacer referencia a un solo archivo. La función `normalizePath` devolverá una ruta de archivo canónica.

Windows tiene los conceptos de nombres de archivo cortos ('8.3') y largos: `normalizePath` devolverá una ruta absoluta con nombres de archivos largos y `shortPathName` devolverá una versión con nombres cortos. Este último no contiene espacios y utiliza la barra invertida como separador, por lo que a veces es útil para exportar nombres de R.

Los permisos de archivos son un tema relacionado. R admite los conceptos POSIX de permiso de lectura / escritura / ejecución para el propietario / grupo / todos, pero es posible que esto solo se admita parcialmente en el sistema de archivos, por lo que, por ejemplo, solo en archivos de solo lectura de Windows (para la cuenta que ejecuta la sesión R) son reconocidos. Las listas de control de acceso (ACL) se emplean en varios sistemas de archivos, pero no tienen un estándar acordado y R no tiene instalaciones para controlarlas. Se usa `Sys.chmod` para cambiar los permisos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Comandos del sistema

Las funciones `system` y `system2` utilizan para invocar un comando del sistema y, opcionalmente, recopilar su salida. `system2` es un poco más general, pero su principal ventaja es que es más fácil escribir código multiplataforma usándolo.

`system` se comporta de manera diferente en Windows desde otros sistemas operativos (porque la llamada a la API C de ese nombre lo hace). En otro lugar, invoca un shell para ejecutar el comando: el puerto de Windows de R tiene una función `shell` para hacerlo.

Para averiguar si el sistema operativo incluye un comando, use `Sys.which`, que intenta hacerlo de forma multiplataforma (desafortunadamente no es un servicio de sistema operativo estándar).

La función `shQuote` citará las rutas de archivo según sea necesario para los comandos en el sistema operativo actual.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Compresión y Archivos.

Las versiones recientes de R tienen amplias instalaciones para leer y escribir archivos comprimidos, a menudo de forma transparente. La lectura de archivos en R se realiza en gran medida mediante conexiones „file” función que se utiliza para abrir una conexión a un archivo (o una URL) y es capaz de identificar la compresión utilizada desde el encabezado del archivo, formato soportado GZIP

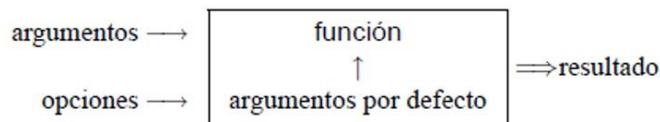
Referencia (<https://en.wikipedia.org/wiki/Xz> y <https://en.wikipedia.org/wiki/LZMA>): R puede leer archivos comprimidos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Como Funciona R

R es un lenguaje de programación orientado a Objetos (significa que las variable, datos, funciones, resultados), son almacenados en la memoria activa, una función de R trabaja de la siguiente forma



El comando más simple es escribir el nombre de un objeto para visualizar su contenido. Por ejemplo, si un objeto n contiene el valor 10:

```
> n  
[1] 10
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Sintaxis y Semántica

- Nombre de los archivos con extensión R ejemplo analisis\_cifras.R
- En los objetos no usar símbolos "-" y "\_" para variables y funciones separar las palabras con "."
- Cada línea de código tendrá un máximo de 80 caracteres, si una línea es muy larga se corta con una ","
- Usar espacios alrededor de todos los operadores binarios (=, +, -, <, entre otros). Los espacios alrededor del símbolo "=" son opcionales cuando se usan para ingresar valores dentro de una función. Así como en español, nunca colocar espacio antes de una coma, pero siempre use espacio luego de una coma.

### Ejemplos

```
tab <- table(df[df$days < 0, 2]) # Bien  
tot <- sum(x[, 1]) # Bien  
tot <- sum(x[, 1]) # Bien  
tab <- table(df[df$days<0, 2]) # Faltan espacios alrededor '<'  
tab <- table(df[df$days < 0,2]) # Falta espacio luego de coma  
tab <- table(df[df$days < 0 , 2]) # Sobra espacio antes de coma  
tab<- table(df[df$days < 0, 2]) # Falta espacio antes de '<'  
tab<-table(df[df$days < 0, 2]) # Falta espacio alrededor de '<'  
tot <- sum(x[,1]) # Falta espacio luego de coma  
tot <- sum(x[1,]) # Falta espacio luego de coma
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Sintaxis y Semántica

- Colocar espacio antes de un paréntesis excepto cuando se llama una función.

Ejemplo

```
if(debug) # Correcto
if(debug) # Funciona pero no se recomienda
colMeans(x) # Funciona pero no se recomienda
```

- Espacios extras pueden ser usados si con esto se mejora la apariencia del código.

Ejemplo

```
plot(x = x.coord,
      y = data.mat[, MakeColName(metric, ptiles[1], "roiOpt")],
      ylim = ylim,
      xlab = "dates",
      ylab = metric,
      main = (paste(metric, " for 3 samples ", sep = "")))
```

- Colocar espacios alrededor del código que esté dentro de paréntesis () o corchetes [ ], la única excepción es luego de una coma,

Ejemplo

```
if(condicion) # Correcto
x[1,] # Correcto
if( condicion ) # Sobran espacios alrededor de condición
x[1,] # Se necesita espacio luego de coma
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Sintaxis y Semántica

- Los signos de agrupación llaves {} se utilizan para agrupar bloques de código y se recomienda que nunca una llave abierta { esté sola en una línea; una llave cerrada } si debe ir sola en su propia línea. Se pueden omitir las llaves cuando el bloque de instrucciones esté formado por una sola línea pero esa línea de código NO debe ir en la misma línea de la condición.

Ejemplo

```
if(is.null(ylim)) { # Correcto
  ylim <- c(0, 0.06)
}

if(is.null(ylim)) { # Correcto
  ylim <- c(0, 0.06)

if(is.null(ylim)) ylim <- c(0, 0.06) # Aceptable

if(is.null(ylim)) { # No se recomienda
{
  ylim <- c(0, 0.06)
}

if(is.null(ylim)) {ylim <- c(0, 0.06)! # Frente a la llave { no debe ir nada
# la llave de cierre } debe ir sola
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Sintaxis y Semántica

- La sentencia else debe ir siempre entre llaves } {

Ejemplo

```
if (condition) {  
  one or more lines  
} else {      # Correcto  
  one or more lines  
}
```

```
if (condition) {  
  one or more lines  
}  
else {      # Incorrecto  
  one or more lines  
}
```

```
if (condition)  
  one line  
else      # Incorrecto  
  one line
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Sintaxis y Semántica

- Asignación

Para realizar asignaciones se recomienda usar el simbolo <- , el simbolo de igualdad = no se recomienda usarlo para asignaciones.

```
x <- 5 # Correcto  
x = 5 # No recomendado
```

- Punto y coma

No se recomienda colocar varias instrucciones separadas por ; en la misma línea, aunque funciona dificulta la revisión del código.

```
n <- 100; y <- rnorm(n, mean=5); hist(y) # No se recomienda
```

```
n <- 100          # Correcto  
y <- rnorm(n, mean=5)  
hist(y)
```

A pesar de la anterior advertencia es posible que en este libro usemos el : en algunas ocasiones, si lo hacemos es para ahorrar espacio en la presentación del código.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Un objeto puede ser creado con el operador “asignar” el cual se denota como una flecha con el signo menos y el símbolo “>” o “<” dependiendo de la dirección en que asigna el objeto:

```
> n <- 15  
> n
```

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> n <- 15  
> n  
[1] 15  
> 5 -> n  
> n  
[1] 5  
> x <- 1  
> X <- 10  
> x  
[1] 1  
> X  
[1] 10  
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Si el objeto ya existe, su valor anterior es borrado después de la asignación (la modificación afecta solo objetos en memoria, no a los datos en el disco).

El valor asignado de esta manera puede ser el resultado de una operación y/o de una función:

```
> n <- 10 + 2  
> n
```

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> n <- 10 + 2  
> n  
[1] 12  
> n <- 3 + rnorm(1)  
> n  
[1] 2.003407  
> |
```

La función `rnorm(1)` genera un dato al azar muestreado de una distribución normal con media 0 y varianza 1

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Escribir una expresión sin asignar su valor a un objeto; en este caso el resultado será visible en la pantalla pero no será guardado en memoria:

```
RGui (64-bit) - [R Console]
> (10 + 2) * 5
[1] 60
```

La asignación será omitida de los ejemplos si no es necesaria para la comprensión del mismo.

La función ls simplemente lista los objetos en memoria; sólo se muestran los nombres de los mismos.

```
RGui (64-bit) - [R Console]
> name <- "oSWALDo"; n1 <- 10; n2 <- 100; m <- 0.5
> ls()
[1] "m"      "n"      "n1"     "n2"     "name"   "x"      "X"
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



El uso del punto y coma para separar comandos diferentes en la misma línea. Si se quiere listar solo aquellos objetos que contengan un carácter en particular, se puede usar la opción pattern (que se puede abbreviar como pat):

```
> ls(pat = "m")
```

Para restringir la lista a aquellos objetos que comienzan con este carácter:

```
RGui (64-bit) - [R Console]
> ls(pat = "^m")
[1] "m"      "n"      "n1"     "n2"     "name"   "x"      "X"

La función ls.str() muestra algunos detalles de los objetos en memoria:

> ls.str()
m : num 0.5
n : num 2
n1 : num 10
n2 : num 100
name : chr "oSWALDo"
x : num 1
X : num 10
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Otra opción útil en esta función es “max.level” que especifica el nivel de detalle de visualización de objetos compuestos. Hay que tener en cuenta que “ls.str( )” muestra todos los detalles de los objetos en memoria, incluyendo las columnas de los marcos de datos (data frames), matrices y listas, lo cual puede generar una gran cantidad de información. Podemos evitar mostrar todos estos detalles con la opción “max.level = -1”.

- rm(nombre) # Borrar "nombre"
- rm(list=ls(pat="x")) # Borra asignaciones que tengan x
- rm(list=ls()) # Borra todo

```
R Gui (64-bit)
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
R Gui (64-bit)
R Console
> nombre<-oSWALDo"; x1<-5; x2<-100; b<-0.5
> B <- data.frame(x1,x2,b)
> ls.str(pat="B")
B : 'data.frame':      1 obs. of  3 variables:
$ x1: num 5
$ x2: num 100
$ b : num 0.5
> rm(nombre) # Borrar "nombre"
> ls()
[1] "b"  "B"  "x1" "x2"
> rm(list=ls(pat="x")) # Borrano asignaciones que tengan x
> ls()
[1] "b"  "B"
> rm(list=ls()) # Borrano todo
> ls()
character(0)
>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



La opción pattern se puede usar de la misma manera con ls.str().

Otra opción útil en esta función es max.level la cual especifica el nivel de detalle para la visualización de objetos compuestos.

Por defecto, ls.str() muestra todos los detalles de los objetos en memoria, incluyendo las columnas de los marcos de datos (“data frames”), matrices y listas, lo cual puede generar una gran cantidad de información. Podemos evitar mostrar todos estos detalles con la opción max.level = -1:

```
> M <- data.frame(n1, n2, m)
> ls.str(pat = "M")
>
> ls.str(pat="M", max.level=-1)
```

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
R Gui (64-bit) - [R Console]
R Gui (64-bit) - [R Console]
> M <- data.frame(n1, n2, m)
> ls.str(pat = "M")
M : 'data.frame':      1 obs. of  3 variables:
$ n1: num 10
$ n2: num 100
$ m : num 0.5
>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com

Para borrar objetos en memoria, utilizamos la función rm(): rm(x) elimina el objeto x, rm(x,y) elimina ambos objetos x y y, y rm(list=ls()) elimina todos los objetos en memoria; las mismas opciones mencionadas para la función ls() se pueden usar para borrar selectivamente algunos objetos: rm(list=ls(pattern="^m")).



La ayuda en línea de R proporciona información muy útil de como utilizar las funciones. La ayuda se encuentra disponible directamente para una función dada.

```
> help("lm")
```

Mostrará dentro de R, ayuda para la función lm() (modelo lineal).

#### Atributos

- Description: descripción breve.
- Usage: para una función, proporciona el nombre de la misma con todos sus argumentos y los posibles valores por defecto (opciones); para un operador describe su uso típico.
- Arguments: para una función, describe en detalle cada uno de sus argumentos.
- Details: descripción detallada.
- Value: si se aplica, el tipo de objeto retornado por la función o el operador.
- See Also: otras páginas de ayuda con funciones u operadores similares.
- Examples: algunos ejemplos que generalmente pueden ser ejecutados sin abrir la ayuda con la función examples().

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



#### Cálculos simples

RGui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> 2+2
[1] 4
> sqrt(9)
[1] 3
> log(3)
[1] 1.098612
> x=5
> y=7
> x+y
[1] 12
> x*y+2
[1] 37
> |
```

#### Concatenaciones

RGui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> x=c(1,2,3,4,5)
> y=c(6,7,8,9,10)
> x
[1] 1 2 3 4 5
> y
[1] 6 7 8 9 10
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejecución de ordenes desde un archivo

### Pasos fundamentales

- Lectura o importación de datos
- Limpieza de formato de datos
- Exploración
- Análisis
- Extracción de Información útil

Los datos planos o (caracteres ASCII) en general que trabajan con caracteres tienen dos peculiaridades:

- El header o encabezamiento
- La separación de caracteres que indican la separación de columnas: pueden estar separadas por comas, puntos y comas, por tabulación, entre otros atributos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Tipo - No tienen encabezado – header – solo datos

```
0.2593466,0,33.25119781,1105.642156,11.25,1,0,16,8.68,1,0,16,0,1.333,0,0
.,-1.54.05338809,2921.768764,.,1,0,9,7.85,1,0,10,1,8,1,0
0.721318058,7,43.57015743,1898.358618,18,1,0,19,8.75,1,0,12,4,3,-1,0
0.011581964,0,30.96783025,959.0065106,16.5,1,1,12,16.31,1,1,12,0,-2,0,1
-0.560984677,0,34.63381246,1199.500965,9.6154,1,1,14,16.85,1,1,14,1,2.917,0,-1
.,2,71.60301164,5126.991275,.,1,0,16,.,1,0,14,-2,24,1,0
1.523260216,2,34.97878166,1223.515166,35,1,0,13,7.63,1,0,15,-2,3,1,0
.,-1,61.45106092,3776.232888,35,1,0,13,.,1,0,14,-2,25,5,0,0
-0.223143551,-2,29.33880904,860.7657156,12,1,1,12,15,1,1,14,1,1,-1,0
-0.470003629,0,47.60574949,2266.307384,6.25,1,0,12,10,1,0,12,0,-5,0,0
0.051751065,-1,51.90143737,2693.759201,17.25,1,1,12,16.38,1,1,13,-2,9,0,0
0.287682073,0,36.07665982,1301.525384,16,1,1,12,12,1,1,12,0,5,0,0
.,4,32.08213552,1029.26342,.,1,0,16,31.25,1,0,12,4,-9.5,0,0
-0.470003629,-2,48.12046544,2315.579194,15.625,1,0,17,25,1,0,19,-3,-7,0,0
0.305601229,2,33.95482546,1152.930172,9.76,1,0,14,7.19,1,0,12,2,.,0,-1
.,4,36.04106776,1298.958565,.,1,0,18,.,1,0,14,4,.,-1,0
-0.336472237,0,27.17590691,738.5299166,1.785714286,1,0,16,2.5,1,0,16,0,0.417,0,0
.,1,24.54483231,602.448793,5.25,1,1,13,.,1,1,12,0,.,1,0
0.628608659,2,39.34017796,1547.649602,15,1,1,14,8,1,1,12,-2,15,1,1
0.321583624,1,38.53251198,1484.754479,15,1,1,13,10.875,1,1,12,-1,0,0,0
.,0,24.65708419,607.9718007,8.7,1,0,12,.,1,0,12,0,-2,0,1
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Tipo - En la primera linea tienes header. A partir de la segunda, datos.

```
PLHRWAGE,DEDUC1,AGE,AGESQ,HRWAGEH,WHITEH,MALEH,EDUCH,HRWAGEL,WHITEL,MALEL,EDUCL,DEDUC2,DTEN,DMARRIED,DUNCOV
0.2593466,0,33.25119781,1105.642156,11.25,1,0,16,8.68,1,0,16,0,1.333,0,0
.,-1,54.05338809,2921.768764,.,1,0,9,7.85,1,0,10,1,8,1,0
0.721318058,7,43.57015743,1898.358618,18,1,0,19,8.75,1,0,12,4,3,-1,0
0.011581964,0,30.96783025,959.0065106,16.5,1,1,12,16.31,1,1,12,0,-2,0,1
-0.560984677,0,34.63381246,1199.500965,9.6154,1,1,14,16.85,1,1,14,1,2.917,0,-1
.,2,71.60301164,5126.991275,.,1,0,16,.,1,0,14,-2,24,1,0
1.523260216,-2,34.97878166,1223.515166,35,1,0,13,7.63,1,0,15,-2,3,1,0
.,-1,61.45106092,3776.232888,35,1,0,13,.,1,0,14,-2,25,5,0,0
-0.223143551,-2,29.33880904,860.7657156,12,1,1,12,15,1,1,14,1,1,-1,0
-0.470003629,0,47.60574949,2266.307384,6.25,1,0,12,10,1,0,12,0,-5,0,0
0.051751065,-1,51.90143737,2693.759201,17.25,1,1,12,16.38,1,1,13,-2,9,0,0
0.287682073,0,36.07665982,1301.525384,16,1,1,12,12,1,1,12,0,5,0,0
.,4,32.08213552,1029.26342,.,1,0,16,31.25,1,0,12,4,-9.5,0,0
-0.470003629,-2,48.12046544,2315.579194,15.625,1,0,17,25,1,0,19,-3,-7,0,0
0.305601229,2,33.95482546,1152.930172,9.76,1,0,14,7.19,1,0,12,2,.,0,-1
.,4,36.04106776,1298.958565,.,1,0,18,.,1,0,14,4,.,-1,0
-0.336472237,0,27.17590691,738.5299166,1.785714286,1,0,16,2.5,1,0,16,0,0.417,0,0
.,1,24.54483231,602.448793,5.25,1,1,13,.,1,1,12,0,.,1,0
0.628608659,2,39.34017796,1547.649602,15,1,1,14,8,1,1,12,-2,15,1,1
0.321583624,1,38.53251198,1484.754479,15,1,1,13,10.875,1,1,12,-1,0,0,0
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Tipo - Encabezado - header - descriptivo

```
=====
KEYWORDS FOR DATASET: Income, Education Level, Twins
=====

=====
ACCOMPANYING DATA PROVIDED BY:
UCLA, Department of Economics
=====

DLHRWAGE,DEDUC1,AGE,AGESQ,HRWAGEH,WHITEH,MALEH,EDUCH,HRWAGEL,WHITEL,MALEL,EDUCL,DEDUC2,DTEN,DMARRIED,DUNCOV
0.2593466,0,33.25119781,1105.642156,11.25,1,0,16,8.68,1,0,16,0,1.333,0,0
.,-1,54.05338809,2921.768764,.,1,0,9,7.85,1,0,10,1,8,1,0
0.721318058,7,43.57015743,1898.358618,18,1,0,19,8.75,1,0,12,4,3,-1,0
0.011581964,0,30.96783025,959.0065106,16.5,1,1,12,16.31,1,1,12,0,-2,0,1
-0.560984677,0,34.63381246,1199.500965,9.6154,1,1,14,16.85,1,1,14,1,2.917,0,-1
.,2,71.60301164,5126.991275,.,1,0,16,.,1,0,14,-2,24,1,0
1.523260216,-2,34.97878166,1223.515166,35,1,0,13,7.63,1,0,15,-2,3,1,0
.,-1,61.45106092,3776.232888,35,1,0,13,.,1,0,14,-2,25,5,0,0
-0.223143551,-2,29.33880904,860.7657156,12,1,1,12,15,1,1,14,1,1,-1,0
-0.470003629,0,47.60574949,2266.307384,6.25,1,0,12,10,1,0,12,0,-5,0,0
0.051751065,-1,51.90143737,2693.759201,17.25,1,1,12,16.38,1,1,13,-2,9,0,0
0.287682073,0,36.07665982,1301.525384,16,1,1,12,12,1,1,12,0,5,0,0
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejecución de ordenes desde un archivo

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[Icons]

> install.packages("readr")
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies 'assertthat', 'utf8', 'cli', 'fansi', 'pillar', 'pkgconfig',
probando la URL 'https://cran.cnr.berkeley.edu/bin/windows/contrib/3.5/assertthat_0.2.0.zip'
Content type 'application/zip' length 53836 bytes (52 KB)
downloaded 52 KB

probando la URL 'https://cran.cnr.berkeley.edu/bin/windows/contrib/3.5/utf8_1.1.4.zip'
Content type 'application/zip' length 214668 bytes (209 KB)
downloaded 209 KB

probando la URL 'https://cran.cnr.berkeley.edu/bin/windows/contrib/3.5/cli_1.0.1.zip'
Content type 'application/zip' length 589901 bytes (576 KB)
downloaded 576 KB

probando la URL 'https://cran.cnr.berkeley.edu/bin/windows/contrib/3.5/fansi_0.4.0.zip'
Content type 'application/zip' length 220351 bytes (215 KB)
downloaded 215 KB

probando la URL 'https://cran.cnr.berkeley.edu/bin/windows/contrib/3.5/pillar_1.3.1.zip'
Content type 'application/zip' length 154999 bytes (151 KB)
downloaded 151 KB

probando la URL 'https://cran.cnr.berkeley.edu/bin/windows/contrib/3.5/pkgconfig_2.0.2.zip'
Content type 'application/zip' length 22157 bytes (21 KB)
downloaded 21 KB

The downloaded binary packages are in
  C:\Users\oswaldo.diaz\AppData\Local\Temp\RtmpYnFSWu\downloaded_packages
[Progress Bar: 6% downloaded]
```

Por la región - selección USA California

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejecución de ordenes desde un archivo

- `read_csv()`: para leer archivos con coma (",") como separador
- `read_csv2()`: para leer archivos con punto y coma (";") como separador
- `read_tsv()`: para leer archivos con tabulador ("\t") como separador
- `read_delim(sep = '|')`: para leer archivos con separador distintos como puede ser el símbolo '|'

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[Icons]

> DatosFuenteSimples <- read_tsv(file.choose())
Parsed with column specification:
cols(
  Nom = col_character(),
  variable = col_double(),
  Group = col_character()
)
> DatosFuenteSimples
# A tibble: 72 x 3
   Nom     variable Group
   <chr>    <dbl> <chr>
 1 IND1      10 A
 2 IND2       7 A
 3 IND3      20 A
 4 IND4      14 A
 5 IND5      14 A
 6 IND6      12 A
 7 IND7      10 A
 8 IND8      23 A
 9 IND9      17 A
10 IND10     20 A
# ... with 62 more rows
> |
```

Cargar los datos

`DatosFuenteSimples <- read_tsv(file.choose())`

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejecución de órdenes desde un archivo

- `read_csv()`: para leer archivos con coma (",") como separador
- `read_csv2()`: para leer archivos con punto y coma (";") como separador
- `read_tsv()`: para leer archivos con tabulador ("\t") como separador
- `read_delim(sep = '|')`: para leer archivos con separador distintos como puede ser el símbolo '|'

DatosInternet <- read\_tsv("http://dominio/ruta/archivo.txt")

### Cargar los datos

DatosFuente <- read\_csv(file.choose())

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Datos <- read\_csv(file.choose(),skip = 0,col\_names = FALSE)

```
> Datos <- read_csv(file.choose(),skip = 0,col_names = FALSE)
> 
> Parsed with column specification:
> cols(
  X1 = col_character(),
  X2 = col_double(),
  X3 = col_double(),
  X4 = col_double(),
  X5 = col_character(),
  X6 = col_double(),
  X7 = col_double(),
  X8 = col_double(),
  X9 = col_character(),
  X10 = col_double(),
  X11 = col_double(),
  X12 = col_double(),
  X13 = col_double(),
  X14 = col_character(),
  X15 = col_double(),
  X16 = col_double()
)
```

```
> Datos
# A tibble: 160 x 16
   X1      X2      X3      X4      X5      X6      X7      X8      X9      X10     X11     X12     X13     X14     X15     X16
   <chr>   <dbl>   <dbl>
 1 0.2593466 0.33.25119781.1105.642156.11.25.1.0.16.8.68.1.0.16.0.1.333.0.0
 2 -1.54.05338809.2921.768764. ,1.0.9.7.85.1.0.16.1.8.1.0
 3 0.72118058.745.37015743.1898.1.0.19.1.5.1.0.12.4.3.1.0
 4 0.41211055.1.0.16.1.8.1.0.16.1.5.1.0.12.4.3.1.0
 5 -0.569984677.0.34.6381246.1199.509965.9.6154.1.1.14.16.85.1.1.14.1.2.917.0.-1
 6 ,2.71.608301164.5126.991275. ,1.0.16.,1.0.14.,2.24.1.0
 7 1.523602002. -2.34.0.16.16.122.0.1.35.1.0.13.0.1.15.2.3.1.8
 8 -1.45.0.16.16.122.0.1.35.1.0.13.0.1.15.2.3.1.8
 9 -0.223143551. -2.29.33889094.860.7657156.12.1.1.12.15.5.1.14.1.1.-1.0
 10 -0.470000269.0.47.68574949.2269.307984.6.25.1.0.12.10.1.0.12.0.5.0.0
 11 0.89312003. -1.0.90143737.269.307984.6.25.1.0.12.10.1.0.12.0.5.0.0
 12 -0.287620774. -1.0.90143737.269.307984.6.25.1.0.12.10.1.0.12.0.5.0.0
 13 ,4.32.08213552.1029.26342.,1.0.16.31.25.1.0.12.4. -9.5.0.0
 14 -0.470000269. -2.48.12046544.2315.579194.15.625.1.0.17.25.1.0.19. -3.7.0.0
 15 0.369000229.2.34.0.16.16.122.0.1.35.1.0.13.0.1.15.2.3.1.8
 16 -0.336472237.0.27.17.959651.738.5299166.1.0.85714286.1.0.16.2.5.1.0.16.0.0.417.0.0
 17 ,1.24.54487231.682.448793.5.25.1.1.13.,1.1.12.0.,-1.0
 18 0.84600059.2.34.0.16.16.122.0.1.35.1.0.13.0.1.15.2.3.1.8
 19 0.321583624.1.38.53231198.1484.7544715.15.11.1.13.10.875.1.1.12.,3.0.0.0
 20 ,0.24.659708439.687.9718807. ,7.1.0.12.,1.0.12.0.,-1.0
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Datos\_encabezado <- read\_csv(file.choose(), skip = 0, col\_names = TRUE)

```

RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[File] [Edit] [View] [Misc] [Help] [Close]

> Datos_encabezado <- read_csv(file.choose(),skip = 0,col_names = TRUE)
Parses with column specification:
cols(
  DLHRWAGE = col_character(),
  DEDUC1 = col_double(),
  AGE = col_double(),
  AGESQ = col_double(),
  HRWAGEH = col_character(),
  WHITEH = col_double(),
  MALEH = col_double(),
  EDUCH = col_double(),
  HRWAGEL = col_character(),
  WHITEL = col_double(),
  MALEL = col_double(),
  EDUCL = col_double(),
  DEDUC2 = col_double(),
  DTEN = col_character(),
  DMARRIED = col_double(),
  DUNCOV = col_double()
)

# A tibble: 160 x 16
#> #>   DLHRWAGE DEDUC1    AGE AGESQ HRWAGEH WHITEH MALEH EDUCH HRWAGEL WHITEL MALEL EDUCL DEDUC2 DTEN
#> #>   <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 0.25934~ 0 33.3 1106. 11.25 1 0 16 8.68 1 0 16 0 16 0 1.333
#> 2 . -1 54.1 2922. . 1 0 9 7.85 1 0 10 1 8
#> 3 0.72131~ 7 43.6 1898. 18 1 0 19 8.75 1 0 12 4 3
#> 4 0.01158~ 0 31.0 959. 16.5 1 1 12 16.31 1 1 12 0 -2
#> 5 -0.5609~ 0 34.6 1200. 9.6154 1 1 14 16.85 1 1 14 1 2.917
#> 6 . 2 71.6 5127. . 1 0 16 . 1 0 14 -2 24
#> 7 1.52326~ -2 35.0 1224. 35 1 0 13 7.63 1 0 15 -2 3
#> 8 . -1 61.5 3776. 35 1 0 13 . 1 0 14 -2 25.5
#> 9 -0.2231~ -2 29.3 861. 12 1 1 12 15 1 1 14 1 1
#> 10 -0.4700~ 0 47.6 2266. 6.25 1 0 12 10 1 0 12 0 -5
#> ... with 150 more rows, and 2 more variables: DMARRIED <dbl>, DUNCOV <dbl>

```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Datos\_encabezado\_resumen <- read\_csv(file.choose(), skip = 8, col\_names = TRUE)

```

RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[File] [Edit] [View] [Misc] [Help] [Close]

> Datos_encabezado_resumen <- read_csv(file.choose(),skip = 8,col_names = TRUE)
Parses with column specification:
cols(
  DLHRWAGE = col_character(),
  DEDUC1 = col_double(),
  AGE = col_double(),
  AGESQ = col_double(),
  HRWAGEH = col_character(),
  WHITEH = col_double(),
  MALEH = col_double(),
  EDUCH = col_double(),
  HRWAGEL = col_character(),
  WHITEL = col_double(),
  MALEL = col_double(),
  EDUCL = col_double(),
  DEDUC2 = col_double(),
  DTEN = col_character(),
  DMARRIED = col_double(),
  DUNCOV = col_double()
)

# A tibble: 160 x 16
#> #>   DLHRWAGE DEDUC1    AGE AGESQ HRWAGEH WHITEH MALEH EDUCH HRWAGEL WHITEL MALEL EDUCL DEDUC2 DTEN
#> #>   <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 0.25934~ 0 33.3 1106. 11.25 1 0 16 8.68 1 0 16 0 16 0 1.333
#> 2 . -1 54.1 2922. . 1 0 9 7.85 1 0 10 1 8
#> 3 0.72131~ 7 43.6 1898. 18 1 0 19 8.75 1 0 12 4 3
#> 4 0.01158~ 0 31.0 959. 16.5 1 1 12 16.31 1 1 12 0 -2
#> 5 -0.5609~ 0 34.6 1200. 9.6154 1 1 14 16.85 1 1 14 1 2.917
#> 6 . 2 71.6 5127. . 1 0 16 . 1 0 14 -2 24
#> 7 1.52326~ -2 35.0 1224. 35 1 0 13 7.63 1 0 15 -2 3
#> 8 . -1 61.5 3776. 35 1 0 13 . 1 0 14 -2 25.5
#> 9 -0.2231~ -2 29.3 861. 12 1 1 12 15 1 1 14 1 1
#> 10 -0.4700~ 0 47.6 2266. 6.25 1 0 12 10 1 0 12 0 -5
#> ... with 150 more rows, and 2 more variables: DMARRIED <dbl>, DUNCOV <dbl>

```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



```
DatosMemoria <- read_csv(file.choose(),skip =  
0,col_names = FALSE)
```

### Analogía

```
> DatosMemoria <- read_csv(file.choose(),skip = 0,col_names = FALSE)  
Parsed with column specification:  
cols(  
  X1 = col_double(),  
  X2 = col_double(),  
  X3 = col_double(),  
  X4 = col_double(),  
  X5 = col_double(),  
  X6 = col_double(),  
  X7 = col_double(),  
  X8 = col_double(),  
  X9 = col_double(),  
  X10 = col_double()  
)  
> DatosMemoria  
# A tibble: 1 × 10  
  X1   X2   X3   X4   X5   X6   X7   X8   X9   X10  
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1  1    2    3    4    5    6    7    8    9    10  
> 1 + DatosMemoria  
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10  
1 2 3 4 5 6 7 8 9 10 11  
> 3.1416 * DatosMemoria  
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10  
1 3.1416 6.2832 9.4248 12.5664 15.709 18.8496 21.9912 25.1328 28.2744 31.416  
> sqrt(DatosMemoria)  
# A tibble: 1 × 10  
  X1   X2   X3   X4   X5   X6   X7   X8   X9   X10  
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1  1  1.41  1.73  2  2.24  2.45  2.65  2.83  3  3.16  
> log(DatosMemoria)  
# A tibble: 1 × 10  
  X1   X2   X3   X4   X5   X6   X7   X8   X9   X10  
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1  0  0.693  1.10  1.39  1.61  1.79  1.95  2.08  2.20  2.30  
>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



```
R Gui (64-bit) - [R Console]
```

```
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
```



```
> x=1024  
> y=2048  
> z=1  
> x+DatosMemoria*y/z  
X1   X2   X3   X4   X5   X6   X7   X8   X9   X10  
1 3072 5120 7168 9216 11264 13312 15360 17408 19456 21504  
> x  
[1] 1024  
> y  
[1] 2048  
> z  
[1] 1  
> DatosMemoria  
# A tibble: 1 × 10  
  X1   X2   X3   X4   X5   X6   X7   X8   X9   X10  
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1  1    2    3    4    5    6    7    8    9    10  
>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



SCAN()

```
Cálculos<-scan("D:\\Software\\R-3.5.3\\datos\\calculadora.txt",sep=",")
```

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> Cálculos<-scan("D:\\Software\\R-3.5.3\\datos\\calculadora.txt",sep=",")  
Read 10 items  
> Cálculos  
[1] 1 2 3 4 5 6 7 8 9 10  
> 3.1416 * Cálculos  
[1] 3.1416 6.2832 9.4248 12.5664 15.7080 18.8496 21.9912 25.1328 28.2744 31.4160  
> ResultadoPI <- 3.1416 * Cálculos  
> ResultadoPI  
[1] 3.1416 6.2832 9.4248 12.5664 15.7080 18.8496 21.9912 25.1328 28.2744 31.4160  
> ResultadoPI / 3.1416  
[1] 1 2 3 4 5 6 7 8 9 10  
> ResultadoEntero <- ResultadoPI / 3.1416  
> ResultadoEntero  
[1] 1 2 3 4 5 6 7 8 9 10  
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Vectores

Los vectores son arreglos ordenados en los cuales se puede almacenar información de tipo numérico (variable cuantitativa), alfanumérico (variable cualitativa) o lógico (TRUE o FALSE), pero no mezclas de éstos. La función de R para crear un vector es c() y que significa concatenar; dentro de los paréntesis de esta función se ubica la información a almacenar. Una vez construido el vector se acostumbra a etiquetarlo con un nombre corto y representativo de la información que almacena, la asignación se hace por medio del operador <- entre el nombre y el vector.

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
presupuesto <- c(1000, 2000, 5000, 10000)
```

```
> presupuesto <- c(1000, 2000, 5000, 10000)  
> presupuesto  
[1] 1000 2000 5000 10000  
> presupuesto_pormes = presupuesto [5] / 12  
> presupuesto_pormes  
[1] NA  
> presupuesto_pormes = presupuesto [3] / 12  
> presupuesto_pormes  
[1] 416.6667  
> presupuesto_pormes = presupuesto [-3] / 12  
> presupuesto_pormes  
[1] 83.33333 166.66667 833.33333  
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Vectores (Manipulación de datos)

Una opción de R es que podemos asignar nombres a los elementos de un vector mediante la función "names"(también se podrá utilizar para nombrar cualquier objeto).

The screenshot shows the RGui (64-bit) interface. The menu bar includes Archivo, Editar, Visualizar, Misc, Paquetes, Ventanas, Ayuda. Below the menu is a toolbar with various icons. The main window is titled 'R Console'. The console output is as follows:

```
> x<-1:7
> names(x)<-c("Lunes","Martes","Miercoles","Jueves","Viernes","Sabado","Domingo")
> x
  Lunes     Martes   Miercoles    Jueves   Viernes   Sabado   Domingo
  1         2          3           4          5          6          7
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Vectores (Manipulación de datos)

Podemos escribir vectores de varias maneras, utilizando la opción ":" (el vector comienza en el primer número suministrado y finaliza en el segundo o en un número anterior sin sobrepasarlo, tanto en orden ascendente como descendente) o mediante la función de concatenación "c()".

- Función "seq(a,b,c)", genera secuencias de números reales, donde el primer elemento indicaría el principio de la secuencia, el segundo el final y el tercero el incremento que se debe usar para generar la secuencia. Aunque también podemos poner la función de estas formas "seq(length = d, from = a, to = b)" o "seq(by = c, from = a, to = b)" siendo "d" la longitud del vector.
- La función "rep(a, b)" que crea un vector con "b" elementos idénticos al valor "a".
- La función "sequence( )" que crea una secuencia de enteros. Cada secuencia terminará en el número especificado como argumento.
- La función "gl( )" (generador de niveles) genera series regulares de factores. Tiene la forma "gl(a, b)" donde "a" es el número de niveles (o clases) y "b" es el número de replicas en cada nivel. Se pueden usar dos opciones: "length" para especificar el número de datos producidos, y "labels" para especificar los nombres de los factores.
- La función "expand.grid( )" crea una tabla de datos con todas las combinaciones posibles de los elementos de los vectores o factores que proporcionemos como argumentos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Matrices

Las matrices son arreglos rectangulares de filas y columnas con información numérica, alfanumérica o lógica. Para construir una matriz se usa la función `matrix()`. Por ejemplo, para crear una matriz de 4 filas y 5 columnas (de dimensión  $4 \times 5$ ) con los primeros 20 números positivos se escribe el código siguiente en la consola

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] 1 5 9 13 17
[2] 2 6 10 14 18
[3] 3 7 11 15 19
[4] 4 8 12 16 20
> mimatriz[3, 4]
[1] 15
> mimatriz[2, ]
[1] 2 6 10 14 18
> mimatriz[, 5]
[1] 17 18 19 20
> mimatriz[, -c(2, 4)]
[1] 1 9 17
[2] 2 10 18
[3] 3 11 19
[4] 4 12 20
> mimatriz[-1, -3]
[1] 2 6 14 18
[2] 3 7 15 19
[3] 4 8 16 20
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Arreglos

Un arreglo es una matriz de varias dimensiones con información numérica, alfanumérica o lógica. Para construir una arreglo se usa la función `array()`. Por ejemplo, para crear un arreglo de  $3 \times 4 \times 2$  con las primeras 24 letras minúsculas del alfabeto se escribe el siguiente código

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
> miarray <- array(data=letters[1:24], dim=c(3, 4, 2))
> miarray
, , 1
[1,] "a" "d" "g" "j"
[2,] "b" "e" "h" "k"
[3,] "c" "f" "i" "l"
, , 2
[1,] "m" "p" "s" "v"
[2,] "n" "q" "t" "w"
[3,] "o" "r" "u" "x"
> miarray[1, 3, 2]
[1] "s"
> miarray[,, 2]
[1,] "m" "p" "s" "v"
[1,] "n" "q" "t" "w"
[2,] "o" "r" "u" "x"
> miarray[, 3]
[1,] "s"
[1,] "g" "s"
[2,] "h" "t"
[3,] "i" "u"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Marco de datos

El marco de datos marco de datos o data frame es uno de los objetos más utilizados porque permite agrupar vectores con información de diferente tipo (numérica, alfanumérica o lógica) en un mismo objeto, la única restricción es que los vectores deben tener la misma longitud. Para crear un marco de datos se usa la función `data.frame()`.

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
> titulo <- c('MSc', 'PhD', 'Preparatoria', 'Secundaria', 'Escuela')
> edad <- c(44, 40, 16, 13, 12)
> nombre <- c('oSWALDO', 'Abel', 'Axel', 'Alan', 'Arath')
> MarcoDatos <- data.frame(nombre, edad, titulo)
> MarcoDatos
  nombre     edad     titulo
1 oSWALDO    44       MSc
2 Abel        40       PhD
3 Axel        16   Preparatoria
4 Alan        13   Secundaria
5 Arath       12       Escuela
> MarcoDatos$titulo
[1] MSc      PhD      Preparatoria Secundaria Escuela
Levels: Escuela MSc PhD Preparatoria Secundaria
> MarcoDatos[, 3]
[1] 44 40 16 13 12
> MarcoDatos[2:4, 1]
[1] Abel Axel Alan
Levels: Abel Alan Arath Axel oSWALDO
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Marco de datos

`subset()`

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> MarcoDatos
  nombre     edad     titulo
1 oSWALDO    44       MSc
2 Abel        40       PhD
3 Axel        16   Preparatoria
4 Alan        13   Secundaria
5 Arath       12       Escuela
> subset(MarcoDatos, subset=edad >= 17)
  nombre     edad     titulo
1 oSWALDO    44       MSc
2 Abel        40       PhD
> subset(MarcoDatos, subset=edad < 20, select=c('nombre', 'titulo'))
  nombre     titulo
3 Axel   Preparatoria
4 Alan   Secundaria
5 Arath   Escuela
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Recolector Datos por internet read.table()

Fuente de datos

[https://raw.githubusercontent.com/fhernanb/datos/master/medidas\\_cuerpo](https://raw.githubusercontent.com/fhernanb/datos/master/medidas_cuerpo)

```
ColeccionDatos <- subset(x=RecolectorDatos, subset=altura > 185 & peso > 80, select=c('sexo', 'edad', 'peso', 'altura'))
```

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] url <- 'https://raw.githubusercontent.com/fhernanb/datos/master/medidas_cuerpo'
[2] RecolectorDatos <- read.table(url, header=T)
[3] dim(RecolectorDatos)
[4] 36 6
[5] ColeccionDatos <- subset(x=RecolectorDatos, subset=altura > 185 & peso > 80, select=c('sexo', 'edad', 'peso', 'altura'))
[6] sexo edad peso altura
[7] 1 Hombre 43 87.3 188.0
[8] 6 Hombre 33 85.9 188.0
[9] 15 Hombre 30 98.2 190.5
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Listas

Objetos que construyen datos de diferentes tipos list()

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] set.seed(12345)
[2] vector <- runif(n=5)
[3] matriz <- matrix(data=1:12, ncol=6)
[4] Lista <- list(Elementos1=vector, Elementos2=matriz, Elementos3=MarcoDatos)
[5] Lista
$Elementos1
[1] 0.7209039 0.8757732 0.7609823 0.8861246 0.4564810

$Elementos2
 [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1     3     5     7     9    11
[2,]    2     4     6     8    10    12

$Elementos3
  nombre edad      titulo
1 oSWALDo   44          MSc
2   Abel    40          PhD
3   Axel   16 Preparatoria
4   Alan   13 Secundaria
5   Arath   12      Escuela
```

La función set.seed de la línea número 1 sirve para fijar la semilla de tal manera que los números aleatorios generados en la segunda linea con la función runif sean siempre los mismos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



list()

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > Lista$Elementos3
[1] > nombre edad titulo
[1] > 1 oSWALDO 44 MSc
[1] > 2 Abel 40 PhD
[1] > 3 Axel 16 Preparatoria
[1] > 4 Alan 13 Secundaria
[1] > 5 Arath 12 Escuela
[1] > Lista[[2]]
[1] > [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 3 5 7 9 11
[2,] 2 4 6 8 10 12
[1] > Lista[2]
[1] $Elementos2
[1] [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 3 5 7 9 11
[2,] 2 4 6 8 10 12
[1] > class(Lista$Elementos2)
[1] "matrix"
[1] > class(Lista[[2]])
[1] "matrix"
[1] > class(Lista[2])
[1] "list"
[1] > Lista$Elementos3
[1] [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 3 5 7 9 11
[2,] 2 4 6 8 10 12
[1] > Lista$Elementos1
[1] 0.7209039 0.8757732 0.7609823 0.8861246 0.4564810
[1] |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Funciones

Las funciones en R se caracterizan por un nombre corto y que dé una idea de lo que hace la función. Los elementos que pueden ingresar (inputs) a la función se llaman parámetros y se ubican dentro de paréntesis, el cuerpo de la función se ubica dentro de llaves y es ahí donde se procesan los inputs para convertirlos en outputs, a continuación se muestra la estructura general de una función.

```
nombre_de_funcion(parametro1, parametro2, ...) {
  tareas internas
  tareas internas
  tareas internas
  salida
}
```

Cuando usamos una función sólo debemos escribir bien el nombre e ingresar correctamente los parámetros de la función, el cuerpo de la función ni lo vemos ni lo debemos modificar.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Funciones sobre vectores

En R podemos destacar las siguientes funciones básicas sobre vectores numéricos.

- min: para obtener el mínimo de un vector.
- max: para obtener el máximo de un vector.
- length: para determinar la longitud de un vector.
- range: para obtener el rango de valores de un vector, entrega el mínimo y máximo.
- sum: entrega la suma de todos los elementos del vector.
- prod: multiplica todos los elementos del vector.
- which.min: nos entrega la posición en donde está el valor mínimo del vector.
- which.max: nos da la posición del valor máximo del vector.
- rev: invierte un vector

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > vector <- c(5, 3, 2, 1, 2, 0, NA, 0, 9, 6)
[1] > min(vector) #no aparece el mínimo que es Cero.
[1] NA
[1] > min(vector, na.rm=TRUE) # Usamos na.rm = TRUE para remover el NA
[1] 0
[1] > max(vector, na.rm=T) # Para obtener el valor máximo
[1] 9
[1] > range(vector, na.rm=T) # Genera min y max simultáneamente
[1] 0 9
[1] > sum(vector, na.rm=T) # La suma de los valores internos
[1] 28
[1] > prod(vector, na.rm=T) # El productor de los valores internos
[1] 0
[1] > which.min(vector) # Posición del valor mínimo 0 en el vector
[1] 6
[1] > which.max(vector) # Posición del valor máximo 9 en el vector
[1] 9
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Función seq

En R podemos crear secuencias de números de una forma sencilla usando la función seq, la estructura de esta función es:

seq(from=1, to=1, by, length.out)  
Los argumentos de esta función son:

- from: valor de inicio de la secuencia.
- to: valor de fin de la secuencia, no siempre se alcanza.
- by: incremento de la secuencia.
- length.out: longitud deseada de la secuencia.

Once valores igualmente espaciados desde 0 hasta 1.  
Una secuencia de dos en dos comenzando en 1.  
Una secuencia desde 1 con un salto de  $\pi$  y sin pasar del número 9.

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > seq(from=0, to=1, length.out = 11)
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
[1] > seq(from=1, to=9, by=2) # matches 'end'
[1] 1 3 5 7 9
[1] > seq(from=1, to=9, by=pi) # stays below 'end'
[1] 1.000000 4.141593 7.283185
[1] > pi:6 # real sequence
[1] 3.141593 4.141593 5.141593
[1] > 6:pi # integer sequence
[1] 6 5 4
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Función rep

En R podemos crear repeticiones usando la función rep, la estructura de esta función es:

```
rep(x, times=1, length.out=NA, each=1)
```

Los argumentos de esta función son:

- x: vector con los elementos a repetir.
- times: número de veces que el vector x se debe repetir.
- length.out: longitud deseada para el vector resultante.
- each: número de veces que cada elemento de x se debe repetir.

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> rep(x=1:4, times=2)
[1] 1 2 3 4 1 2 3 4
> rep(x=1:4, times=c(2,2,2,2))
[1] 1 1 2 2 3 3 4 4
> rep(x=1:4, times=c(2,1,2,1))
[1] 1 1 2 3 3 4
> rep(x=1:4, each=2)
[1] 1 1 2 2 3 3 4 4
> |
```

• 1 2 3 4 1 2 3 4  
• 1 1 2 2 3 3 4 4  
• 1 2 3 3 4  
• 1 1 2 2 3 3 4 4

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Funciones round, ceiling, floor y trunc

Existen 4 funciones útiles para modificar u obtener información de un número, estas funciones son round, ceiling, floor y trunc.

- round(x, digits): sirve para redondear un número según los dígitos indicados.
- ceiling(x): entrega el mínimo entero mayor o igual que x.
- floor(x): entrega el máximo entero menor o igual que x.
- trunc(x): entrega la parte entera de un número x

funciones round, ceiling, floor y trunc a un valor positivo y a un valor negativo para inspeccionar los resultados.

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> x <- 5.34896 # Número positivo elegido
> round(x, digits=3)
[1] 5.349
> ceiling(x)
[1] 6
> floor(x)
[1] 5
> trunc(x)
[1] 5
> x <- -4.26589 # Número negativo elegido
> round(x, digits=3)
[1] -4.266
> ceiling(x)
[1] -4
> floor(x)
[1] -5
> trunc(x)
[1] -4
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Funciones sort y Rank

Las funciones sort y rank son útiles para ordenar los elementos de un vector o para saber las posiciones que ocuparían los elementos de un vector al ser ordenado. La estructura de las dos funciones es la siguiente.

- `sort(x, decreasing = FALSE)`
- `rank(x)`

En el parámetro `x` se ingresa el vector y el parámetro `decreasing` sirva para indicar si el ordenamiento es de menor a mayor (por defecto es este) o de mayor a menor.

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> x <- c(2, 3, 6, 4, 9, 5)
> sort(x)
[1] 2 3 4 5 6 9
> sort(x, decreasing=TRUE)
[1] 9 6 5 4 3 2
> rank(x)
[1] 1 2 5 3 6 4
>
```

Considere el vector `x` que tiene los siguientes elementos: 2, 3, 6, 4, 9 y 5. Ordene el vector de menor a mayor, de mayor a menor y por último encuentre la posición que ocupan los elementos de `x` si se ordenaran de menor a mayor.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Funciones algebraicas

Función	Operación
<code>+, -, *, /</code>	Suma, Resta, Multiplicación, División
<code>abs</code>	Valor absoluto
<code>asin acos atan</code>	Inversas de las funciones trigonométricas
<code>exp, log</code>	Exponencial y logaritmo natural
<code>round</code>	Redondeo
<code>sin cos tan</code>	Funciones trigonométricas
<code>sqrt(), ^</code>	Raíz cuadrada, Potencia
<code>%/%</code>	División entera
<code>%%</code>	Resto de la división

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Expresiones lógicas	
<, <=	Menor que; Menor o igual que
>, >=	Mayor que; Mayor o igual que
==	Igual a
!=	No igual a
Operadores lógicos	
&	“y”
	“o”
!	“no”

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Función	Salida
length()	Longitud del vector
sum()	Suma de los elementos del vector
prod()	Producto de los elementos del vector
max()	Máximo valor del vector
min()	Mínimo valor del vector
cumsum()	Vector de sumas acumulada de los elementos
cummax()	Vector de máximos acumulados
cummin()	Vector de mínimos acumulados
cumprod()	Producto acumulado de los elementos
diff()	Vector de diferencias entre elementos
unique()	Un vector de valores únicos
duplicated()	Vector lógico que indica si los elementos están duplicados

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Funciones Estadísticas

Función	Resultado
median ()	Mediana del vector <b>x</b>
mean()	Media del vector
quantile()	Cuantiles
IQR()	Rango intercuartil
range (x)	Rango del vector <b>x</b>
sd (x)	Desviación estándar
var()	Varianza de los elementos
summary()	Resumen descriptivos

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Calculo Matricial

Función	Salida
chol()	Descomposición de Cholesky
col()	Matriz cuyos elementos son el número de columna
det()	Determinante de la matriz
diag()	Extrae los elementos de la diagonal de <b>x</b>
eigen()	Computa Eigenvalues y Eigenvectores
ncol()	Devuelve el número de columnas de la matriz
nrow()	Devuelve el número de filas de la matriz
qr()	Descomposición qr de una matriz
row()	Matriz cuyos elementos son el número de fila
solve()	Calcula la inversa de <b>x</b>
svd()	Descomposición en valores singulares
t()	Transpuesta de la matriz <b>x</b>
var()	Estima la matriz de varianzas-covarianzas
cor()	Estima la matriz de correlaciones
%%	Producto matricial

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Función	Resultado
<code>colnames()</code>	Nombres de las columnas de la matriz
<code>colSums()</code>	Suma los elementos de las columnas
<code>dim()</code>	Dimensiones de la matriz
<code>length()</code>	Número de elementos de la matriz
<code>Rownames</code>	Nombres de las filas de la matriz
<code>rowSums()</code>	Suma los elementos de las filas de una matriz
<code>colMeans()</code>	Obtiene la media aritmética de cada columna
<code>rowMeans()</code>	Obtiene la media aritmética de cada fila

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Aritmética Vectorial

Los vectores pueden ser manipulados en forma similar a los escalares utilizando las mismas funciones que presentamos en la última sección. Sin embargo, se debe ser muy cuidadoso/a al sumar o restar vectores de diferentes longitudes ya que se pueden obtener resultados inesperados.

Nombre	Operación
<code>length()</code>	devuelve la cantidad de componentes del vector
<code>sum()</code>	calcula la suma de las componentes del vector
<code>prod()</code>	calcula el producto de las componentes del vector
<code>cumsum()</code> , <code>cumprod()</code>	calcula sumas y productos acumulados
<code>sort()</code>	ordena el vector
<code>diff()</code>	calcula diferencias de vectores adecuadamente corridos (por defecto 1)

R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

<img icons>

```
> x <- c(1,2,3,4)
> y <- c(5,6,7,8)
> x*y
[1]  5 12 21 32
> y/x
[1] 5.000000 3.000000 2.333333 2.000000
> y-x
[1] 4 4 4 4
> x^y
[1]      1    64  2187 65536
> cos(x*pi) + cos(y*pi)
[1] -2  2 -2  2
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Generación de sucesiones

En R existen varias funciones para generar sucesiones numéricas. Por ejemplo, 1:30 es el vector c(1,2,...,29,30). El operador 'dos puntos' tiene máxima prioridad en una expresión, así, por ejemplo, 2\*1:15 es el vector c(2,4,6,...,28,30). Escriba n <- 10 y compare las sucesiones 1:n-1 y 1:(n-1).

La forma 30:1 permite construir una sucesión descendente.

La función seq() permite generar sucesiones más complejas. Dispone de cinco argumentos, aunque no se utilizan todos simultáneamente. Si se dan los dos primeros indican el comienzo y el final de la sucesión, y si son los únicos argumentos, el resultado coincide con el operador 'dos puntos', esto es, seq(2,10) coincide con 2:10.

Los argumentos de seq(), y de muchas funciones de R, pueden darse además de por posición, por nombre, en cuyo caso, el orden en que aparecen es irrelevante.

En esta función los dos primeros argumentos se pueden dar por nombre mediante from=valor-inicial y to=valor-final; por tanto seq(1,30), seq(from=1, to=30) y seq(to=30, from=1) son formas equivalentes a 1:30.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Generación de sucesiones

Los dos siguientes argumentos de seq() son by=valor y length=valor, y especifican el 'paso' y 'longitud' de la sucesión respectivamente. Si no se suministra ninguno, el valor predeterminado es by=1 y length se calcula.

### Ejemplo

```
> seq(-5, 5, by=.2) -> s3  
genera el vector c(-5.0, -4.8, -4.6, ..., 4.6, 4.8, 5.0) y lo almacena en s3 . Similarmente
```

```
> s4 <- seq(length=51, from=-5, by=.2)  
genera los mismos valores y los almacena en s4.
```

El quinto argumento de esta función es along=vector, y si se usa debe ser el único argumento, ya que crea una sucesión 1, 2, ..., length(vector), o la sucesión vacía si el vector es vacío (lo que puede ocurrir).

Una función relacionada con seq es rep(), que permite duplicar un objeto de formas diversas. Su forma más sencilla es

```
> s5 <- rep(x, times=5)  
que coloca cinco copias de x, una tras otra, y las almacena en s5.
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Vectores de índices. Selección y modificación de subvectores

Puede seleccionar un subvector de un vector añadiendo al nombre del mismo un vector de índices entre corchetes, [ y ]. En general podrá obtener un subvector de cualquier expresión cuyo resultado sea un vector, sin más que añadirle un vector de 'índices' entre corchetes.

Los vectores de 'índices' pueden ser de cuatro tipos distintos:

Un vector lógico. En este caso el vector de 'índices' debe tener la misma longitud que el vector al que refiere. Solo se seleccionaran los elementos correspondientes a valores T del vector de índices y se omitirá el resto.

Por ejemplo,

```
> y <- x[!is.na(x)]
```

almacena en y los valores no-faltantes de x, en el mismo orden. Si x tiene valores faltantes, el vector y será más corto que x. Análogamente,

```
> (x+1)[(!is.na(x)) & x>0] -> z
```

almacena en z los elementos del vector x+1 para los que el correspondiente elemento de x es no-faltante y positivo.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Vectores de índices. Selección y modificación de subvectores

Un vector de números naturales positivos. En este caso los elementos del vector de índices deben pertenecer al conjunto {1, 2, . . . , length(x)}.

El resultado es un vector formado por los elementos del vector referido que corresponden a estos índices y en el orden en que aparecen en el vector de índices. El vector de índices puede tener cualquier longitud y el resultado será de esa misma longitud. Por ejemplo, x[6] es el sexto elemento de x, y

```
> x[1:10]
```

es el vector formado por los diez primeros elementos de x, (supuesto que length(x) no es menor que 10). Por otra parte,> c("x","y")[rep(c(1,2,2,1), times=4)] crea un vector de caracteres de longitud 16 formado por "x", "y", "y", "x" repetido cuatro veces.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Vectores de índices. Selección y modificación de subvectores

Un vector de números naturales negativos. En este caso, los índices indican los elementos del vector referido que deben excluirse. Así pues,

```
> y <- x[-(1:5)]  
almacena en el vector y todos los elementos de x excepto los cinco primeros (suponiendo que x tiene al menos cinco elementos).
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Vectores de índices. Selección y modificación de subvectores

Un vector de caracteres. Esta opción solo puede realizarse si el vector posee el atributo names (nombres) para identificar sus componentes, en cuyo caso se comportará de modo similar al punto 2.

```
> fruta <- c(5, 10, 1, 20)  
> names(fruta) <- c("naranja", "platano", "manzana", "pera")  
> postre <- fruta[c("manzana", "naranja")]
```

La ventaja en este caso es que los nombres son a menudo más fáciles de recordar que los índices numéricos. Esta opción es especialmente útil al tratar de la estructura de “hoja de datos” (data frame) que veremos posteriormente.

La variable de almacenamiento también puede ser indexada, en cuyo caso la asignación se realiza solamente sobre los elementos referidos. La expresión debe ser de la forma vector[vector de índices] ya que la utilización de una expresión arbitraria en vez del nombre de un vector no tiene mucho sentido.

El vector asignado debe ser de la misma longitud que el vector de índices y, en el caso de un vector de indices lógico, debe ser de la misma longitud del vector que indexa. Por ejemplo,

```
> x[is.na(x)] <- 0  
sustituye cada valor faltante de x por un cero. Por otra parte,  
> y[y < 0] <- -y[y < 0]  
equivale a  
> y <- abs(y)
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

Datos del PGA Tour 2010-2018

Todas las estadísticas registradas en el PGA Tour desde la temporada 2010 hasta la 2018.



Contexto: Conjunto de datos de las estadísticas de todos los jugadores en el PGA Tour de la temporada 2010 a 2018.

R Gui (64-bit) - [R Console]

R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> library("readr")
> DatosFuente <- read_csv(file.choose(), skip = 0, col_names = FALSE)
Parsed with column specification:
cols(
  X1 = col_character(),
  X2 = col_character(),
  X3 = col_character(),
  X4 = col_character(),
  X5 = col_character()
)
=====
> dim(DatosFuente)
[1] 2591063      5
>
226 MB = 2'591,063 datos<
```

<https://www.kaggle.com/datasets?sortBy=hotness&groupBy=public&page=1&pageSize=20&size=large&filetype=all&license=all>

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

Validar el diccionario de datos [Lógica del negocio]

Columnas

- [X1] Nombre del jugador: Nombre del deportista
- [X2] Temporada: Temporada de interés
- [X3] Estadística Métrica de rendimiento
- [X4] Variable: Subcategorías dentro de cada estadística
- [X5] Valor de cada subcategoría dentro de cada estadística.



Parsed with column specification:
cols(
 X1 = col\_character(),
 X2 = col\_character(),
 X3 = col\_character(),
 X4 = col\_character(),
 X5 = col\_character()

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

Crear un agrupamiento de datos para analizarlo (solo el año 2018)



306,799 universo de datos

R Gui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> subset (DatosFuente, subset=X2 == 2018, select=c(X1, X2, X3, X4, X5))
# A tibble: 306,799 x 5
  X1          X2          X3          X4          X5
  <chr>       <chr>       <chr>       <chr>       <chr>
1 Rory McIlroy 2018 Driving Distance Driving Distance - (ROUNDS) 64
2 Trey Mullinax 2018 Driving Distance Driving Distance - (ROUNDS) 75
3 Tom Lovelady 2018 Driving Distance Driving Distance - (ROUNDS) 87
4 Tony Finau   2018 Driving Distance Driving Distance - (ROUNDS) 105
5 Luke List    2018 Driving Distance Driving Distance - (ROUNDS) 98
6 Dustin Johnson 2018 Driving Distance Driving Distance - (ROUNDS) 77
7 Gary Woodland 2018 Driving Distance Driving Distance - (ROUNDS) 96
8 Brooks Koepka 2018 Driving Distance Driving Distance - (ROUNDS) 64
9 Bubba Watson 2018 Driving Distance Driving Distance - (ROUNDS) 90
10 Keith Mitchell 2018 Driving Distance Driving Distance - (ROUNDS) 99
# ... with 306,789 more rows
> dim(DatosFuente)
[1] 2591063      5
> ColeccionDatos2018 <- subset(x=DatosFuente, subset=X2 == 2018, select=c(X1, X2, X5))
> dim(ColeccionDatos2018)
[1] 306799       3
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

- Descriptores

[X3] Estadística Métrica de rendimiento

Victory Leaders: Lideres diez veces en torneos del año 2018"



R Gui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> Lideres2018 <- subset(x=DatosFuente, subset=X3 == 'Victory Leaders' & X2 == 2018 & X5 == 10, select=c(X1, X3, X5))
> Lideres2018
# A tibble: 9 x 3
  X1          X3          X5
  <chr>       <chr>       <chr>
1 Tommy Gainey Victory Leaders 10
2 Colt Knost   Victory Leaders 10
3 Rick Lamb    Victory Leaders 10
4 George McNeill Victory Leaders 10
5 John Peterson Victory Leaders 10
6 Carl Pettersson Victory Leaders 10
7 Vijay Singh   Victory Leaders 10
8 Julian Suri   Victory Leaders 10
9 Omar Uresti   Victory Leaders 10
> dim(Lideres2018)
[1] 9 3
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejemplo

- Descriptores

[X3] Estadística Métrica de rendimiento

Eagles (Holes per): Resultado que se obtiene al jugar un hoyo en dos golpes menos de su par. mas de 50 veces por año.



R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> HolesPer <- subset(x=DatosFuente, subset=X3 == 'Eagles (Holes per)' & X2 == 2018 & X5 > 50 , select=c(X1, X3, X5))
> dim(HolesPer)
[1] 285 3
> HolesPer
# A tibble: 285 x 3
   X1          X3          X5
   <chr>        <chr>        <chr>
 1 Justin Thomas Eagles (Holes per) 91
 2 Dustin Johnson Eagles (Holes per) 77
 3 Conrad Shindler Eagles (Holes per) 60
 4 Webb Simpson    Eagles (Holes per) 95
 5 Hideki Matsuyama Eagles (Holes per) 75
 6 Keegan Bradley  Eagles (Holes per) 98
 7 Stewart Cink    Eagles (Holes per) 87
 8 Matt Every      Eagles (Holes per) 79
 9 Cameron Percy   Eagles (Holes per) 56
10 Rafa Cabrera Bello Eagles (Holes per) 75
# ... with 275 more rows
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejemplo

- Descriptores

[X3] Estadística Métrica de rendimiento

Total Eagles: Jugadores que en dos golpes embocaron la bola mas de 50 veces por año.



R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> TotalEagles <- subset(x=DatosFuente, subset=X3 == 'Total Eagles' & X2 == 2018 & X5 > 50 , select=c(X1, X3, X5))
> dim(TotalEagles)
[1] 260 3
> TotalEagles
# A tibble: 260 x 3
   X1          X3          X5
   <chr>        <chr>        <chr>
 1 Webb Simpson Total Eagles 95
 2 Justin Thomas Total Eagles 91
 3 Keegan Bradley Total Eagles 98
 4 Dustin Johnson Total Eagles 77
 5 Stewart Cink   Total Eagles 87
 6 Jason Kokrak   Total Eagles 99
 7 Hideki Matsuyama Total Eagles 75
 8 Gary Woodland Total Eagles 96
 9 Bryson DeChambeau Total Eagles 93
10 Matt Every     Total Eagles 79
# ... with 250 more rows
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejemplo

- Descriptores

[X3] Estadística Métrica de rendimiento

Total Birdies: Resultado que se obtiene al jugar un hoyo en un golpe menos de su par, mas de 50 veces por año.



R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> TotalBirdies <- subset(x=DatosFuente, subset=X3 == 'Total Birdies' & X2 == 2018 & X5 > 50, select=c(X1, X3, X5))
> dim(TotalBirdies)
[1] 175   3
> TotalBirdies
# A tibble: 175 x 3
   X1          X3          X5
   <chr>        <chr>        <chr>
 1 Keith Mitchell Total Birdies 99
 2 Bryson DeChambeau Total Birdies 93
 3 Jason Kokrak    Total Birdies 99
 4 Phil Mickelson Total Birdies 90
 5 Luke List      Total Birdies 98
 6 Aaron Wise     Total Birdies 90
 7 Emiliano Grillo Total Birdies 94
 8 Marc Leishman   Total Birdies 93
 9 Justin Thomas   Total Birdies 91
10 Xander Schauffele Total Birdies 96
# ... with 165 more rows
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejemplo

Tiger Woods



R Gui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



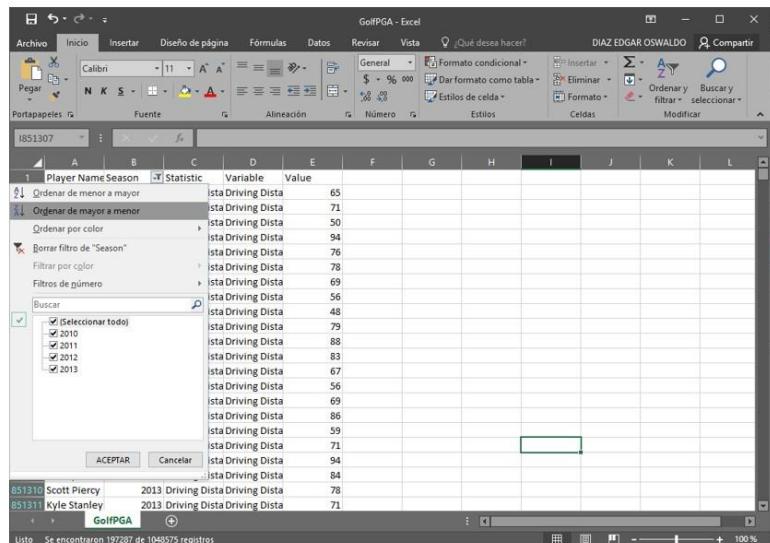
```
> Lider  <- subset(x=DatosFuente, subset=X1 == 'Victory Leaders' & X2 == 2018, select=c(X4, X5))
> HolesPer <- subset(x=DatosFuente, subset=X1 == 'Tiger Woods' & X3 == 'Eagles (Holes per)' & X2 == 2018, select=c(X4, X5))
> TotalEagles <- subset(x=DatosFuente, subset=X1 == 'Tiger Woods' & X3 == 'Total Eagles' & X2 == 2018, select=c(X4, X5))
> TotalBirdies <- subset(x=DatosFuente, subset=X1 == 'Tiger Woods' & X3 == 'Total Birdies' & X2 == 2018, select=c(X4, X5))
> ResumenTigerWoods <- data.frame(Lider, HolesPer, TotalEagles, TotalBirdies)
> ResumenTigerWoods
   X4 X5          X4.1 X5.1          X4.2 X5.2          X4.3 X5.3
1  Victory Leaders - (EVENTS) 18    Eagles (Holes per) - (ROUNDS)  68 Total Eagles - (ROUNDS)  68
2 Victory Leaders - (VICTORIES) 1    Eagles (Holes per) - (FREQUENCY) 204.0 Total Eagles - (TOTAL)      6 Total Birdies - (TOTAL) 274
3  Victory Leaders - (EVENTS) 18    Eagles (Holes per) - (# OF HOLES) 1,224 Total Eagles - (ROUNDS)  68 Total Birdies - (ROUNDS)  68
4  Victory Leaders - (VICTORIES) 1    Eagles (Holes per) - (# OF EAGLES)       6 Total Eagles - (TOTAL)      6 Total Birdies - (TOTAL) 274
> |
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Clases – Definición

Los objetos básicos en R tienen una clase implícita definida en S3. Es accesible con class.

R Gui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda  
  
> Actividad <- list(what='Mandar un correo',  
+ when=as.Date('2019-04-06'), priority='Media')  
> class(Actividad) <- 'Curso de R UCA'  
> Actividad  
\$what  
[1] "Mandar un correo"  
  
\$when  
[1] "2019-04-06"  
  
\$priority  
[1] "Media"  
  
attr("class")  
[1] "Curso de R UCA"  
> |

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Clases – Definición

Los objetos básicos en R tienen una clase implícita definida en S3. Es accesible con class.

R Gui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda  
  
> Actividad2 <- list(what='Entrega Proyecto R UCA',  
+ when=as.Date('2019-04-06'), priority='Alta')  
> class(Actividad2) <- 'Horario 13 horas'  
> Actividad2  
\$what  
[1] "Entrega Proyecto R UCA"  
  
\$when  
[1] "2019-04-06"  
  
\$priority  
[1] "Alta"  
  
attr("class")  
[1] "Horario 13 horas"  
> |

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Clases – Definición

Los objetos básicos en R tienen una clase implícita definida en S3. Es accesible con class.

```
R Gui (64-bit) - [R Console]
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > Agenda <- list(Actividad, Actividad2)
[2] > class(Agenda) <- c('Actividades')
[3] > Agenda
[4] > 
[5] > $when
[6] > "Mandar un correo"
[7] > 
[8] > $when
[9] > "2019-04-06"
[10] > $priority
[11] > "Media"
[12] > attr(,"class")
[13] > "Curso de R UCA"
[14] > 
[15] > $when
[16] > "Entrega Proyecto R UCA"
[17] > 
[18] > $when
[19] > "2019-04-06"
[20] > $priority
[21] > "Alta"
[22] > attr(,"class")
[23] > "Horario 13 horas"
[24] > attr(,"class")
[25] > "Actividades"
[26] > |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Métodos

UseMethod sirve para elegir el método correspondiente a la clase del objeto empleado como argumento en la función. Se debe definir un método genérico, incluyendo llamada a UseMethod.

```
R Gui (64-bit) - [R Console]
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > miFunc <- function(x, ...)UseMethod('miFunc')
[2] > miFunc.default <- function(x, ...){
[3] + cat('Función genérica\n')
[4] + print(x)
[5] + }
[6] > x <- rnorm(10) # distribución normal
[7] > miFunc(x)
Función genérica
[1] -0.850589091  0.115835043  0.002266247 -0.517694247 -0.478071283  0.135211764  1.282762567  0.017432898
[9]  0.398455934  0.832263232
[10] > |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Métodos

```
RGui (64-bit) - [R Console]
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] <img> [2] <img> [3] <img> [4] <img> [5] <img> [6] <img> [7] <img> [8] <img> [9] <img> [10] <img>

> Actividad
$what
[1] "Mandar un correo"

$when
[1] "2019-04-06"

$priority
[1] "Media"

attr(),"class")
[1] "Curso de R UCA"
> miFunc(Actividad)
Funcion genérica
$what
[1] "Mandar un correo"

$when
[1] "2019-04-06"

$priority
[1] "Media"

attr(),"class")
[1] "Curso de R UCA"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Métodos

Con methods() podemos averiguar los métodos que hay definidos para una función particular

```
RGui (64-bit) - [R Console]
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] <img> [2] <img> [3] <img> [4] <img> [5] <img> [6] <img> [7] <img> [8] <img> [9] <img> [10] <img>

> methods('miFunc')
[1] miFunc.default
see '?methods' for accessing help and source code
> methods('summary')
[1] summary.aov           summary.aoqvlist*      summary.aspell*
[4] summary.check_packages_in_dir* summary.connection  summary.data.frame
[7] summary.Date          summary.default       summary.ecdf*
[10] summary.factor        summary.glm          summary.infl*
[13] summary.lm            summary.loess*       summary.manova
[16] summary.matrix        summary.nlm*        summary.nls*
[19] summary.packageStatus* summary.PDF_Dictionary* summary.PDF_Stream*
[22] summary.POSIXct        summary.POSIXlt       summary.ppr*
[25] summary.prcomp*       summary.princomp*    summary.proc_time
[28] summary.srcref         summary.srcref      summary.stepfun
[31] summary.stl*          summary.table       summary.tukeysmooth*
[34] summary.warnings       see '?methods' for accessing help and source code
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Métodos Ejemplo

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[|] R Gui (64-bit) - [R Console]
[|] Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[|] miFunc.Actividad <- function(x, numero,...)
+ {
if (!missing(numero))
cat('Actividad número', numero,':\n')
cat('Que realizar: ', x$what,
'- Cuando:', as.character(x$when),
'- Prioridad:', x$priority,
'\n')
}

miFunc(Actividad)
methods(miFunc)
methods(class='Actividad')

print.Actividad <- function(x, ...){
cat('Actividad:\n')
NextMethod(x, ...) ## Ejecuta print.default
}

print(Actividad)
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Métodos Ejemplo

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[|] R Gui (64-bit) - [R Console]
[|] Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[|] print.Agenda3 <- function(x, ...){
+ cat('Lista de Actividades:\n')
+ NextMethod(x, ...)
+ cat('-----\n')
+ }

> print(Agenda)
Lista de Actividades:
[1]
$what
[1] "Mandar un correo"

$when
[1] "2019-04-06"

$priority
[1] "Media"

attr(,"class")
[1] "Curso de R UCA"

[2]
$what
[1] "Entrega Proyecto R UCA"

$when
[1] "2019-04-06"

$priority
[1] "Alta"

attr(,"class")
[1] "Horario 13 horas"

attr(,"class")
[1] "Actividades"
-----
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Métodos Ejemplo

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > print(Agenda[[2]])
$what
[1] "Entrega Proyecto R UCA"
$when
[1] "2019-04-06"
$priority
[1] "Alta"
attr(),"class")
[1] "Horario 13 horas"
> print(Agenda[[1]])
$what
[1] "Mandar un correo"
$when
[1] "2019-04-06"
$priority
[1] "Media"
attr(),"class")
[1] "Curso de R UCA"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Métodos Ejemplo

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > print.Agenda <- function(x, ...){
+ cat('Agenda Lista de actividades:\n')
+ for (i in seq_along(x)){
+ cat('Número de tarea:', i,'|\n')
+ cat('Que actividad:', x[[i]]$what,
+ '- Cuando:', as.character(x[[i]]$when),
+ '- Prioridad:', x[[i]]$priority,
+ '\n')
+ }
+ cat('-----\n')
+ }
> print(Agenda)
Lista de Actividades:
[[1]]
$what
[1] "Mandar un correo"
$when
[1] "2019-04-06"
$priority
[1] "Media"
attr(),"class")
[1] "Curso de R UCA"
[[2]]
$what
[1] "Entrega Proyecto R UCA"
$when
[1] "2019-04-06"
$priority
[1] "Alta"
attr(),"class")
[1] "Horario 13 horas"
```

Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



```
RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

> print.Agenda <- function(x, ...){
+   cat('Agenda: Lista de actividades:\n')
+   for (i in seq_along(x)) print(x[[i]], i)
+   cat('-----\n')
+ }
> print(Agenda)
Lista de Actividades:
[1]
$what
[1] "Mandar un correo"
$when
[1] "2019-04-06"
$priority
[1] "Media"
attr("class")
[1] "Curso de R UCA"
[2]
$what
[1] "Entrega Proyecto R UCA"
$when
[1] "2019-04-06"
$priority
[1] "Alta"
attr("class")
[1] "Horario 13 horas"
attr("class")
[1] "Actividades"
-----
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



- sapply y lapply (que son casi la misma)
- tapply
- apply y mapply
- Las funciones ddply, ldply, entre otras. del paquete plyr

```
RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

> Datos <- function(x) if(x < 5) x^2 else -x^2
> sapply(1:10, Datos)
[1] 1 4 9 16 -25 -36 -49 -64 -81 -100
>
> lapply(iris, class)
$Sepal.Length
[1] "numeric"

$Sepal.Width
[1] "numeric"

$Petal.Length
[1] "numeric"

$Petal.Width
[1] "numeric"

$Species
[1] "factor"

> sapply(iris, length)
Sepal.Length Sepal.Width Petal.Length Petal.Width      Species
          150           150           150           150           150
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Programación funcional en R

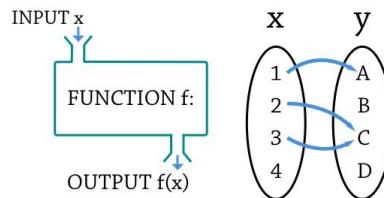
Funciones anónimas

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> sapply(1:10, function(Datos) if(Datos < 5) Datos^2 else -Datos^2)
[1]  1   4   9  16 -25 -36 -49 -64 -81 -100
> |
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Map, reduce

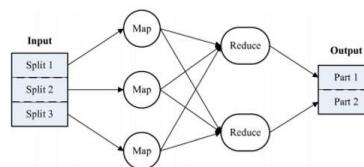
Existen dos operaciones fundamentales en programación funcional. La primera es map y consiste en aplicar una función a todos los elementos de una lista o vector.

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> sapply(1:10, function(Datos) if(Datos < 5) Datos^2 else -Datos^2)
[1]  1   4   9  16 -25 -36 -49 -64 -81 -100
>
> sqrt(1:10)
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427 3.000000 3.162278
>
> Reduce(function(a, b) a + b, 1:10)
[1] 55
>
> sum(1:10)
[1] 55
> sum(sqrt(1:10))
[1] 22.46828
> |
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



R está orientado a objetos. Los objetos son estructuras que combinan datos y funciones que operan sobre ellos y son muy útiles en un entorno, como R, pensado para el análisis estadístico de datos.

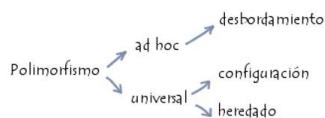
Por ejemplo, el resultado de una regresión lineal es un objeto (de clase lm):

```
R Gui (64-bit) - [R Console]
> Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] "RGui"
> str(mi.modelo)
List of 12
$ coefficients : Named num [1:2] -17.58 3.93
..- attr(*, "names")= chr [1:2] "(Intercept)" "speed"
$ residuals   : Named num [1:50] 3.85 11.85 -5.95 12.05 2.12 ...
..- attr(*, "names")= chr [1:50] "1" "2" "3" "4" ...
$ effects     : Named num [1:50] -303.914 145.551 -8.115 9.885 0.194 ...
..- attr(*, "names")= chr [1:50] "(Intercept)" "speed" " " ...
$ rank        : int 2
$ fitted.values: Named num [1:50] -1.85 -1.85 9.95 9.95 13.88 ...
..- attr(*, "names")= chr [1:50] "1" "2" "3" "4" ...
$ assign       : int [1:2] 0 1
$ qr          : List of 5
..$ qr       : num [1:50, 1:2] -7.071 0.141 0.141 0.141 0.141 ...
... ..- attr(*, "dimnames")=List of 2
... ... $ : chr [1:50] "1" "2" "3" "4" ...
... ... $ : chr [1:2] "(Intercept)" "speed"
... ..- attr(*, "assign")= int [1:2] 0 1
..$ qraux: num [1:2] 1.14 1.27
..$ pivot: int [1:2] 1 2
..$ tol : num 1e-07
..$ rank : int 2
..- attr(*, "class")= chr "qr"
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



El polimorfismo es una característica del lenguaje que se logra en R gracias a la orientación a objetos, permite que una única función, opere de manera distinta dependiendo de su argumento.



```
R Gui (64-bit) - [R Console]
> Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
```

```
> is.list(mi.modelo)
[1] TRUE
> summary(mi.modelo)

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-29.069 -9.525 -2.272  9.215 43.201 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -17.5791     6.7584  -2.601  0.0123 *  
speed         3.9324     0.4155   9.464 1.49e-12 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438 
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

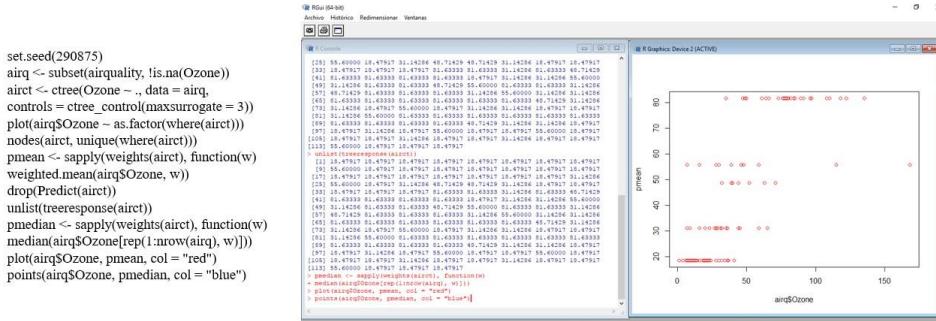
(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Arboles de decisiones

Los árboles de decisión son útiles para entender la estructura de un conjunto de datos. Sirven para resolver problemas tanto de clasificación (predecir una variable discreta, típicamente binaria) como de regresión (predecir una variable continua).

```
install.packages("party")
library(party)
```



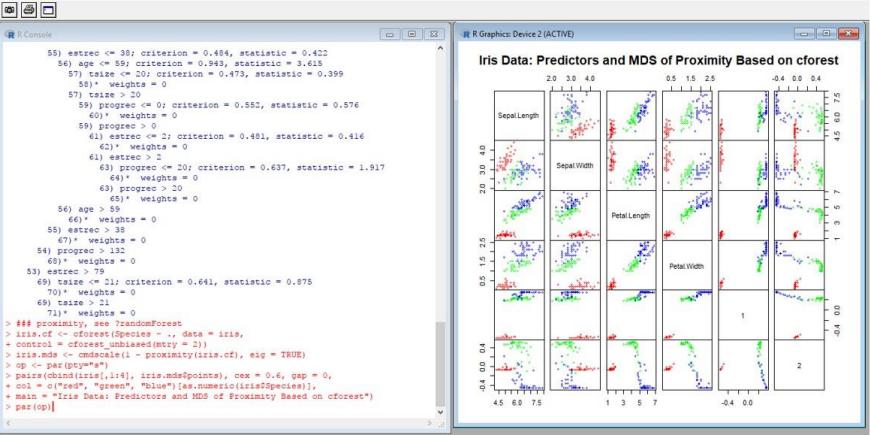
<https://www.statmethods.net/advstats/cart.html>

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Arboles de decisiones

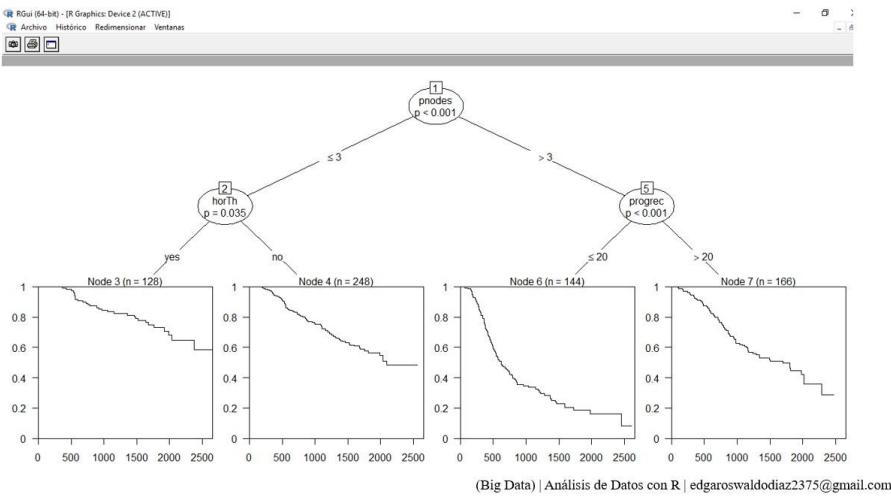
```
R Gui (64-bit)
Archivo Histórico Redimensionar Ventanas
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Arboles de decisiones



### Que es S3

Los conceptos de clase y método son fundamentales para cualquier sistema orientado a objetos. Una clase define un tipo de objeto, describe las propiedades que posee, cómo se comporta y cómo se relaciona con otros tipos de objetos. Cada objeto debe ser una instancia de alguna clase. Un método es una función asociada con un tipo particular de objeto.

S3 implementa un estilo de programación orientada a objetos llamada OO de función genérica. Esto es diferente a la mayoría de los lenguajes de programación, como Java, C ++ y C #, que implementan OO de paso de mensajes. En el estilo de paso de mensajes, los mensajes (métodos) se envían a los objetos y el objeto determina a qué función llamar. Por lo general, este objeto tiene una apariencia especial en la llamada al método, generalmente aparece antes del nombre del método / mensaje: por ejemplo `canvas.drawRect("blue")`. S3 es diferente. Mientras que los cálculos todavía se realizan a través de métodos, un tipo especial de función llamada función genérica decide a qué método llamar. Los métodos se definen de la misma manera que una función normal, pero se llaman de una manera diferente, como veremos en breve.

El uso principal de la programación OO en R es para los métodos de impresión, resumen y trazado. Estos métodos nos permiten tener una función genérica, por ejemplo `print()`, que muestra el objeto de manera diferente según su tipo: la impresión de un modelo lineal es muy diferente a la impresión de un marco de datos.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Clases en S4

Se construyen con `setClass`, que acepta varios argumentos

- `class`: nombre de la clase.
- `slots`: una lista con las clases de cada componente. Los nombres de este vector corresponden a los nombres de los componentes (`slot`).
- `contains`: un vector con las clases que esta nueva clase extiende.
- `prototype`: un objeto proporcionando el contenido por defecto para los componentes definidos en `slots`.
- `validity`: una función que comprueba la validez de la clase creada con la información suministrada

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[Icons]

> setClass('Actividad',
+   slots = c(
+     what = 'character',
+     when = 'Date',
+     priority = 'character'
+   )
+ )
> getClass('Actividad')
Class "Actividad" [in ".GlobalEnv"]

Slots:

Name:           what      when    priority
Class: character Date character
> getSlots('Actividad')
  what      when    priority
"character" "Date" "character"
> slotNames('Actividad')
[1] "what"    "when"   "priority"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

Una vez que la clase ha sido definida con `setClass`, se puede crear un objeto nuevo con `new`.

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[Icons]

> Actividad3 <- new('Actividad',
+   what="Reporte de calificaciones",
+   when=as.Date('2019-04-08'),
+   priority='High')
>
> Actividad3
An object of class "Actividad"
Slot "what":
[1] "Reporte de calificaciones"

Slot "when":
[1] "2019-04-08"

Slot "priority":
[1] "High"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

```
> createActividad <- function(what, when, priority){\n+ new('Actividad',\n+ what = what,\n+ when = when,\n+ priority = priority)\n+ }\n>\n> Actividad4 <- createActividad(what = 'Entregar Reporte de actividades',\n+ when = as.Date('2019-03-30'),\n+ priority = 'Media')\n> Actividad4\nAn object of class "Actividad"\nSlot "what":\n[1] "Entregar Reporte de actividades"\n\nSlot "when":\n[1] "2019-03-30"\n\nSlot "priority":\n[1] "Media"\n> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

```
> setClass('Agenda',\n+ slots = c(Actividad = 'list')\n+ )\n>\n> ListaActividades <- new('Agenda',\n+ Actividad = list(\n+ a3 = Actividad3,\n+ a4 = Actividad4))\n> ListaActividades\nAn object of class "Agenda"\nSlot "Actividad":\n[2] \nActividad:\nAn object of class "Actividad"\nSlot "what":\n[1] "Reporte de calificaciones"\n\nSlot "when":\n[1] "2019-04-08"\n\nSlot "priority":\n[1] "High"\n\n\na4\nActividad:\nAn object of class "Actividad"\nSlot "what":\n[1] "Entregar Reporte de actividades"\n\nSlot "when":\n[1] "2019-03-30"\n\nSlot "priority":\n[1] "Media"
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejemplo

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

> ListaActividades@Actividad
$@3
Actividad:
An object of class "Actividad"
Slot "what":
[1] "Reporte de calificaciones"

Slot "when":
[1] "2019-04-08"           ListaActividades@Actividad
                           ListaActividades@Actividad$@1
                           ListaActividades@Actividad$@2@what

Slot "priority":
[1] "High"                 $@4
                           Actividad:
                           An object of class "Actividad"
                           Slot "what":
                           [1] "Entregar Reporte de actividades"

Slot "when":
[1] "2019-03-30"           $@4
                           Actividad:
                           An object of class "Actividad"
                           Slot "what":
                           [1] "Entregar Reporte de actividades"

Slot "priority":
[1] "Media"
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Estructuras de control y manejo de datos

En los lenguajes de programación, se entiende por estructuras de control aquellas construcciones sintácticas del lenguaje que dirigen el flujo de la ejecución de un programa en una dirección o en otra dentro del código. Por ejemplo, prácticamente todos los lenguajes tienen una construcción "IF", que permite ejecutar o saltar un conjunto, bloque o secuencia de instrucciones dentro del código de un programa. R también cuenta con un conjunto de estructuras de control, si bien, mucho de lo que éstas implementan se puede también hacer mediante

### La construcciones IF-ELSE

Estas construcciones son semejantes a las de otros lenguajes de programación, con una salvedad que puede ser capitalizada por los usuarios del lenguaje: la construcción en sí misma regresa un valor, que puede, si se quiere, ser asignado a una variable o utilizado de otras maneras. Los siguientes ejemplos muestran la sintaxis y el uso de estas construcciones

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

> aa <- 15
> if (aa > 14) # if sin else
+ print("SI MAYOR")
[1] "SI MAYOR"
>
> if (aa > 14) print ("SI MAYOR")
[1] "SI MAYOR"
>
> if (aa > 14) { # Instrucción compuesta
+ print ("PRIMER RENGLON")
+ print ("SI MAYOR")
+
[1] "PRIMER RENGLON"
[1] "SI MAYOR"
>
>
> y <- 10
> y <- if (aa > 14) 50
> y
[1] 50
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



La construcción IF admite una sola sola expresión, pero ésta puede ser la expresión compuesta, que se construye mediante los paréntesis de llave {}, y las expresiones en su interior, separadas ya sea por el cambio de renglón o por ;'. En los casos anteriores, la expresión señalada por el if se ejecuta u omite dependiendo del valor de la condición, TRUE o FALSE, respectivamente. En el caso del ejemplo la condición está dada por la expresión aa > 14, que prueba si la variable aa es mayor que 14. La siguientes construcciones, redirigen la ejecución del código a distintos bloques o conjuntos de instrucciones dependiendo de que se cumplan o no las condiciones establecidas:

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > if (10 > aa) { # 1er. bloque
+ print("RANGO MENOR")
+ } else if ( 10 <= aa && aa <= 20) { # 2o. bloque
+ print("primer renglon"); print("RANGO MEDIO")
+ } else { # 3er. bloque
+ print("RANGO MAYOR")
+
[1] "primer renglon"
[1] "RANGO MEDIO"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



#### Repeticiones por un número determinado de veces

La construcción que habilita esta operación es la instrucción for. El número de veces que se repite la expresión o expresiones englobadas en la instrucción, puede estar explícita en ella misma.

El número de veces que se repite la expresión, puede quedar implícito en las estructuras de las cuales toma elementos el for. Así, el mismo resultado que se obtuvo en el ejemplo anterior.

```
R Gui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] > letras <- c("C", "I", "A", "M", "T", "R")
[2] > for (i in 1:6) {
+ print(letras[i])
+
[1] "C"
[1] "I"
[1] "A"
[1] "M"
[1] "T"
[1] "R"
>
[3] > for (i in seq_along(letras)) {
+ print(letras[i])
+
[1] "C"
[1] "I"
[1] "A"
[1] "M"
[1] "T"
[1] "R"
>
[4] > for (letra in letras) {
+ print(letra)
+
[1] "C"
[1] "I"
[1] "A"
[1] "M"
[1] "T"
[1] "R"
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Ejemplo

R Gui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda  
 > # Repeticiones mientras se cumple una condición  
> i <- 1  
> while (i <= 6) {  
+ print(letras[i])  
+ i <- i + 1  
+ }  
[1] "c"  
[1] "i"  
[1] "z"  
[1] "h"  
[1] "t"  
[1] "a"  
>  
> # Repeticiones infinitas  
> i <- 1  
> repeat {  
+ print(letras[i])  
+ i <- i + 1  
+ if (i > 6)  
+ break  
+ }  
[1] "c"  
[1] "i"  
[1] "z"  
[1] "h"  
[1] "t"  
[1] "a"  
> |

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Ejemplo

R Gui (64-bit) - [R Console]  
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda  
 > # Interrupciones del flujo normal de los ciclos  
> set.seed(140) # el argumento puede ser cualquier número  
> aprox <- 0.003 # Valor determinante para la salida del ciclo  
> Y\_ini <- 2.7 # Supuesto valor inicial de Y  
> for (iter in 1:1000) { # aseguro no más de 1000 iteraciones  
+ # Procedimiento para calcular la siguiente Y, que  
+ # en este caso simularemos mediante generador aleatorio:  
+ Y <- Y\_ini + 0.008\*rnorm(1)  
+ # La condición de salida:  
+ if (abs(Y - Y\_ini) <= aprox)  
+ break # Uso del break para salir del ciclo  
+ # Preparamos para la siguiente iteración  
+ Y\_ini <- Y  
+ }  
> # Veamos que ha resultado:  
> paste0("Y\_ini: ", Y\_ini, ", Y: ", Y, ", Num.iter: ", iter)  
[1] "Y\_ini: 2.76443400590741, Y: 2.76582777768031, Num.iter: 8"  
> |

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ciclos con Funciones

```
RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] <--> <--> <--> <--> <--> <--> <--> <-->

> # Primero se crea la función:
> fibonacci <- function(n) {
+ if (n %in% c(0,1))
+ return (1)
+ F0 <- 1; F1 <- 1; i <- 2
+ repeat {
+ s <- F0 + F1 # Suma de los fib anteriores
+ if (i == n) # Ya es el que se busca
+ return (s) # Sale hasta afuera de la función
+ # recorremos los últimos dos próximos números
+ F0 <- F1
+ F1 <- s
+ i <- i+1 # incrementamos el indice
+ }
+ }
> # El octavo número de fibonacci se genera
> # llamando a la función así:
> fibonacci(8)
[1] 34
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Ejemplo

```
RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[1] <--> <--> <--> <--> <--> <--> <--> <-->

> # Recursividad
> MiFact <- function(n) {
+ if (n==0) return (1) # salida inmediata
+ if (n > 0) return (n*MiFact(n-1))
+ return (NULL) # caso fallido
+ }
> MiFact(5)
[1] 120
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Cambiar Datos cuantitativos a cualitativos

R Gui (64-bit)

Archivo Ventanas Editar Ayuda

	V1	V2	V3	V4	v
1	1	3	3	1	
2	2	3	3	2	
3	3	3	3	2	
4	1	3	3	1	
5					
6					
7					
8					
9					
10					
11					
12					

> summary(Datos[1])

V1

Min. :1.00  
1st Qu.:1.00  
Median :1.50  
Mean :1.75  
3rd Qu.:2.25  
Max. :3.00

> |

R Console

```
> Datos = read.table("clipboard")
> Datos
  V1 V2 V3 V4
1  1  3  3  1
2  2  3  3  2
3  3  3  3  2
4  1  3  3  1
> fix(Datos)
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Cambiar Datos cuantitativos a cualitativos

R Gui (64-bit)

Archivo Ventanas Editar Ayuda

	V1	V2	V3	V4	va
1	numero uno1	3	3	1	
2	numero uno2	3	3	2	
3	numero uno3	3	3	2	
4	numero uno1	3	3	1	
5					
6					
7					
8					
9					
10					
11					
12					

R Console

```
> Datos <- transform(Datos,
+ V1=factor(V1, labels=c("numero uno")))
> Datos
  V1 V2 V3 V4
1 numero uno1 3 3 1
2 numero uno2 3 3 2
3 numero uno3 3 3 2
4 numero uno1 3 3 1
> summary(Datos[1])
  V1
numero uno1:1
numero uno2:1
numero uno3:1
> fix(Datos)
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Cambiar Datos cuantitativos a cualitativos

```
R Gui (64-bit) - [R Console]
  Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
  [File Edit View Packages Windows Help]
  > summary(Datos[2,3,4])
    Min. 1st Qu. Median   Mean 3rd Qu.   Max.
      3          3       3       3       3       3
  > summary(Datos[2])
    V2
    Min. 1st Qu. Median   Mean 3rd Qu.   Max.
      3          3       3       3       3       3
  > summary(Datos[3])
    V3
    Min. 1st Qu. Median   Mean 3rd Qu.   Max.
      3          3       3       3       3       3
  > summary(Datos[4])
    V4
    Min. 1st Qu. Median   Mean 3rd Qu.   Max.
      1.0      1.0      1.5     1.5     2.0     2.0
  > |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Renombrar Variables

```
R Gui (64-bit) - [R Console]
  Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
  [File Edit View Packages Windows Help]
  > Datos = read.table("clipboard")
  > Datos
    V1 V2 V3 V4
  1  1  3  3  1
  2  2  3  3  2
  3  3  3  3  2
  4  1  3  3  1
  > names (Datos) [c(3)] <- c ("Argumento")
  > Datos
    V1 V2 Argumento V4
  1  1  3           3  1
  2  2  3           3  2
  3  3  3           3  2
  4  1  3           3  1
  > |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Crear nuevas variables a partir de las existentes

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> Datos
  V1 V2 Argumento V4
1 1 3      3 1
2 2 3      3 2
3 3 3      3 2
4 1 3      3 1
> Datos$LGV2 <- with(Datos, log(V2))
> Datos
  V1 V2 Argumento V4      LGV2
1 1 3      3 1 1.098612
2 2 3      3 2 1.098612
3 3 3      3 2 1.098612
4 1 3      3 1 1.098612
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Análisis Univariado | Frecuencias

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda



```
> Datos
  V1 V2 V3 V4
1 1 3 3 1
2 2 3 3 2
3 3 3 3 2
4 1 3 3 1
> attach(Datos)
The following objects are masked from Datos (pos = 3):
  V1, V2, V3, V4
> table(V2)
V2
3
4
> cuadrol <- table(V2)
> cuadrol
V2
3
4
> cuadrol/margin.table(cuadrol) # Frecuencia relativa
V2
3
1
> cumsum(round(((cuadrol/margin.table(cuadrol))*100),2)) # Frecuencia relativa acumulada
  3
100
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



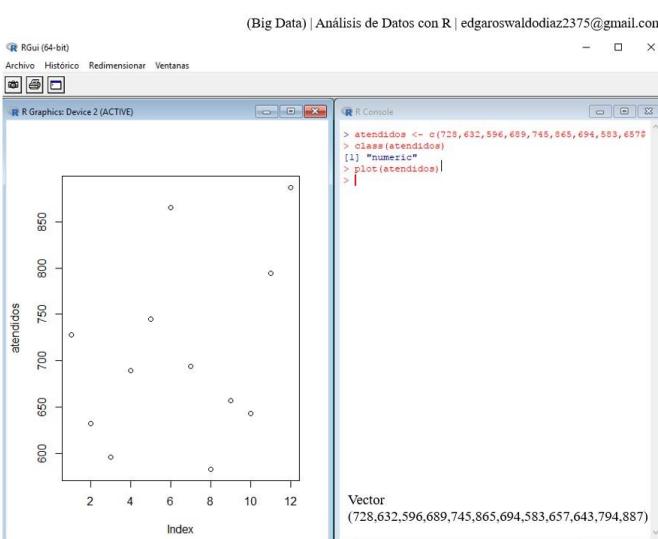
```
RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
[Icons]

> summary(Datos)
      V1          V2          V3          V4
Min. :1.00  Min. :3  Min. :3  Min. :1.0
1st Qu.:1.00 1st Qu.:3 1st Qu.:3 1st Qu.:1.0
Median :1.50 Median :3  Median :3  Median :1.5
Mean   :1.75 Mean   :3  Mean   :3  Mean   :1.5
3rd Qu.:2.25 3rd Qu.:3 3rd Qu.:3 3rd Qu.:2.0
Max.  :3.00 Max.  :3  Max.  :3  Max.  :2.0
> summary(Datos[, "V4"])
      Min. 1st Qu. Median  Mean 3rd Qu.  Max.
1.0    1.0    1.5    1.5    2.0    2.0
> |
```



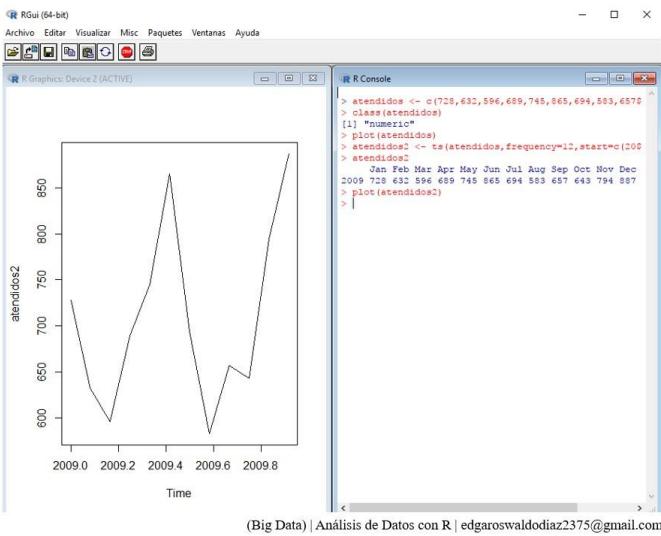
En términos genéricos, todos los elementos que maneja R son objetos: un valor numérico es un objeto, un vector es un objeto, una función es un objeto, una base de datos es un objeto, un gráfico es un objeto, para realizar un uso eficiente de R es preciso entender y aprender a manipular bien las distintas clases de objetos que maneja el programa. En esta sección nos vamos a ocupar particularmente de aquellos objetos que R utiliza para representar datos: valores, vectores, matrices, dataframes, series temporales y listas.

El vector representa el número de personas atendidas mensualmente en el servicio de urgencias de un centro de salud durante el pasado año (datos de enero a diciembre)





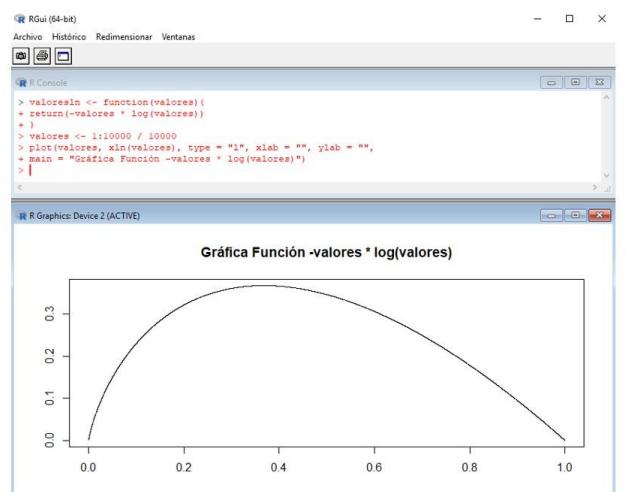
La función `plot()` como muchas otras funciones en R, primero comprueba cuál es la clase del objeto sobre el que debe actuar, y en función de ello se comporta de una manera u otra.



Estos datos en serie temporal mediante la función `ts()`, indicando que esta serie comienza en enero del año 2009 y que tiene una frecuencia de 12 observaciones por año (esto es, una por mes)



Las expresiones condicionales permiten ejecutar un código u otro en función de un criterio. Para ilustrar su uso en R, Comenzaremos definiendo y representando gráficamente la función `xln`, que aparece frecuentemente en estadística.

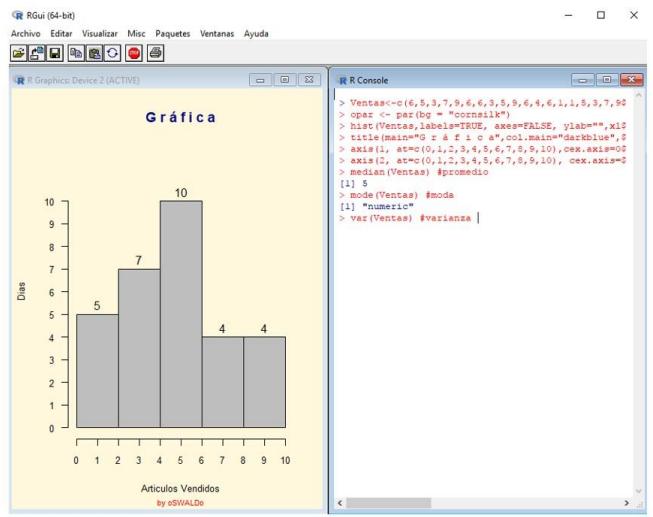


(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Graficando Datos

```
Ventas<-  
c(6,5,3,7,9,6,6,3,5,9,6,4,6,1,1,5,3,7,9,2,1,3,5,9,5,  
4,8,7,0,3)  
opar <- par(bg = "cornsilk")  
hist(Ventas,labels=TRUE, axes=FALSE,  
ylab="",xlab="",main="",col="gray",ylim=c(0,1  
2))  
title(main="G r á f i c  
a",col.main="darkblue",xlab="Articulos  
Vendidos",ylab="Dias",col.lab="black",cex.lab=0.80,  
sub="by oSWALDo",  
col.sub="red",cex.sub=0.7)  
axis(1, at=c(0,1,2,3,4,5,6,7,8,9,10),cex.axis=0.8)  
axis(2, at=c(0,1,2,3,4,5,6,7,8,9,10), cex.axis=0.8,  
las=2)  
median(Ventas) #promedio  
mode(Ventas) #moda  
var(Ventas) #varianza
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

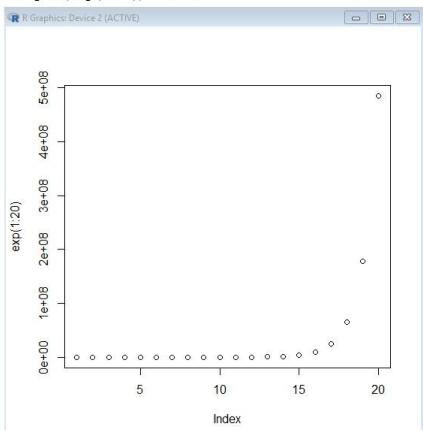
- abline( ) Función que añade una o mas líneas rectas.
- plot( ) Función genérica para representar en el plano xy puntos, líneas, etc.
- barplot( ) Diagramas de barras.
- pie( ) Diagramas de sectores.
- hist( ) Histogramas.
- boxplot( ) Diagramas de box-and-whisker.
- stripplot( ) Similares a boxplot( ) con puntos.
- sunflowerplot( ) Representación en el plano xy de diagramas de girasol.
- qqnor( ) Diagramas de cuantil a cuantil frente a la distribución normal.
- qqplot( ) Diagramas de cuantil a cuantil de dos muestras.
- qline( ) Representa la línea que pasa por el primer y el tercer cuartil.
- lines( ) Añade líneas a un gráfico.
- points( ) Añade puntos a un gráfico.
- segments( ) Añade segmentos a un gráfico.
- arrows( ) Añade flechas a un gráfico.
- polygons( ) Añade polígonos a un gráfico.
- rect( ) Añade rectángulos a un gráfico.
- abline( ) Añade una recta de pendiente e intersección dada.
- curve( ) Representa una función dada.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com

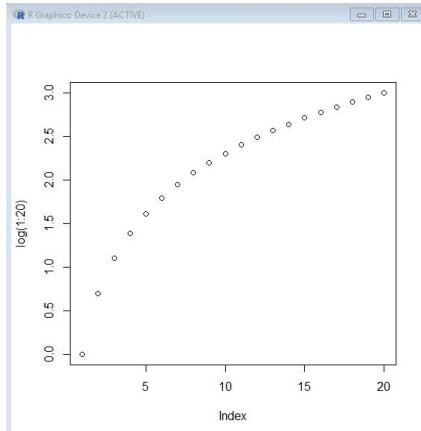


Plot

```
plot(exp(1:20))
```



```
plot(log(1:20))
```

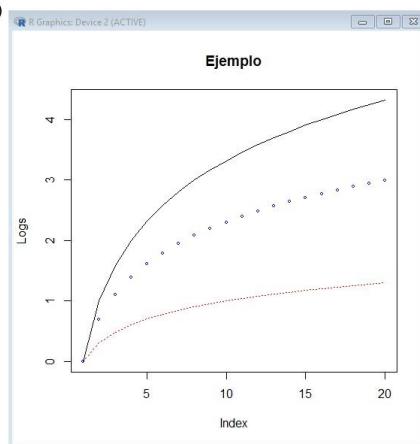


(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Plot

```
plot(log2(1:20), type="l", main="Ejemplo", ylab="Logs")
points(log(1:20), col="blue", cex=0.5)
lines(log10(1:20), col="red", lty=3)
```

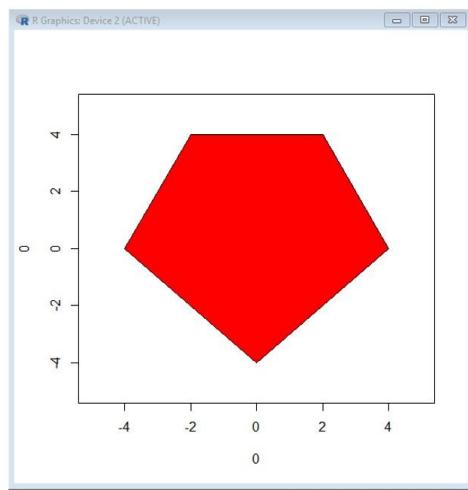


(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Ejemplo

```
plot(0,0,type="n",xlim=c(-5,5),ylim=c(-5,5))
x <-c(0,4,2,-2,-4)
y <-c(-4,0,4,4,0)
polygon(x,y,col="red",border="black")
```

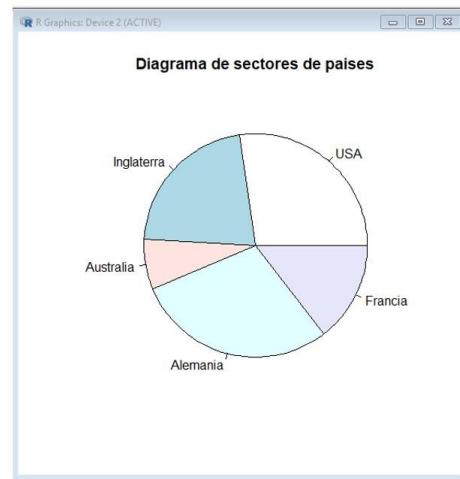


(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Pie

```
muestra <- c(15, 12,4, 16, 8)
paises<- c("USA", "Inglaterra", "Australia",
"Alemania", "Francia")
pie(muestra, labels = paises, main="Diagrama de
sectores de paises")
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



plotrix

```
install.packages("plotrix")
library(plotrix)

escanos <- c(125, 120, 4, 16, 8)
partidos <- c("PSOE", "PP", "IU", "CIU", "PNV")
pie3D(escanos,labels=partidos,explode=0.1, main="Elecciones Nacionales")
```

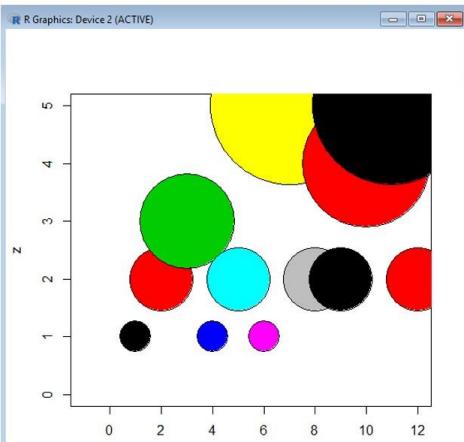


(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



symbols

```
x=1:12
z<-c(1,2,3,1,2,1,5,2,2,4,5,2)
symbols(x,z,circles=z,xlim=c(-1,12),ylim=c(0,
5),bg=1:10)
```



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Funciones	Formatos gráficos
Dispositivos pantalla	
x11() o X11	Ventana x de Windows
windows()	Microsoft Windows
quartz()	Mac OS X Quartz
Archivos	
postscript()	Adobe postscript
pdf()	Adobe PDF
pictex()	Latex Pictex
xfig()	XFIG
bitmap()	GhostScript
png()	PNG
jpeg()	JPEG
Sólo para Windows	
win.metafile()	Windows Metafile
bmp()	Windows BMF

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Función	Tipo de gráfico
plot()	Barras, dispersión, cajas
hist()	Histograma
stem.leaf()	Tallo y hojas
boxplot()	Gráfico de cajas
qqplot()	Gráfico QQ de cuantiles
scatterplot	Diagrama de dispersión
pie()	Gráfico de sectores
dotchart(x)	Gráfico Cleveland

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Argumentos gráficos	Efecto
<code>type = ""</code>	Especifica el tipo de gráfico: "p" puntos "l" líneas "p" puntos y líneas "n" No gráfico "o" puntos y líneas "h" líneas verticales
<code>axes =</code>	F elimina los ejes T dibuja los ejes (defecto)
<code>xlab = "texto"</code>	Añade texto al eje
<code>ylab = "texto"</code>	Título del gráfico
<code>main = "texto"</code>	Subtítulo bajo el eje X
<code>sub = "texto"</code>	Escalas de los ejes X e Y
<code>xlim = c(mínimo, máximo)</code>	
<code>ylim = c(mínimo, máximo)</code>	

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



#### Funciones graficas de bajo nivel

Funciones de bajo nivel	Efecto
<code>points (x,y,...)</code>	Añade puntos a la función gráfica (plot) inmediatamente anterior. El tipo de punto se fija con pch = y su tamaño por cex = o mkh =
<code>lines(x,y,...)</code>	Añade líneas a la función gráfica (plot) inmediatamente anterior. El tipo de linea se fija con lty = y su tamaño por lwd =
<code>text(x,y,labels,...)</code>	Añade texto a un gráfico en los puntos de coordenadas x,y
<code>abline(a,b,h = ,v = ...)</code>	Añade una recta con intercepto (a) y pendiente (b). h = fija el valor de y para las líneas horizontales. v = fija el valor de X para las líneas verticales
<code>legend (x,y, text...)</code>	Añade al gráfico una leyenda en la posición especificada. Por defecto las leyendas se imprimen dentro de un rectángulo.
<code>title(main = "texto", sub = "texto", xlab = "texto", ylab = "texto", line = NA, outer = FALSE, ...)</code>	Añade texto a un gráfico. Un título principal que por defecto aparecerá en la parte superior, un subtítulo (abajo el eje X), y etiquetas para los ejes X e Y.
<code>axis (side,labels, tick...)</code>	side = 1 eje inferior; 2 = eje izquierdo; 3 = eje superior; 4 = eje derecho. Sus argumentos permiten controlar las marcas de división (tick) y las etiquetas de división (lab)
<code>segments(x0,y0,x1,y1)</code>	Dibuja segmentos en las coordenadas dadas
<code>polygon(x,y-)</code>	Añade polígonos cerrados
<code>arrows(x0,y0,x1,y1)</code>	Añade flechas
<code>symbols (x,y,...)</code>	Dibuja círculos, cuadrados....

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Parámetros gráficos

Parámetros gráficos	Efecto
<code>mfrow = c(m,n)</code>	Permite dibujar m filas y n columnas de gráficos
<code>mfg = c(i,j)</code>	Especifica las coordenadas en las que se dibujará el gráfico
<code>ask = TRUE</code>	Se pregunta al usuario antes de borrar un gráfico para dibujar uno nuevo
<code>mar = c(s1,s2,s3,s4)</code>	Fija los márgenes del plot
<code>oma = c(s1,s2,s3,s4)</code>	Fija los márgenes externos

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## Parámetros para los elementos gráficos

Elementos gráficos	Efecto
<code>pch = " "</code>	Define el tipo de carácter. El usuario puede seleccionar entre 19 caracteres diferentes a cada uno de los cuales le corresponde un indicador numérico entre 0 y 18
<code>cex =</code>	Especifica el tamaño de los caracteres. Recibe el nombre de carácter de expansión.
<code>mkh =</code>	Especifica el tamaño de las marcas de los ejes
<code>lth =</code>	Tipo de linea, entre 1 (línea continua) y 6.
<code>lwd =</code>	Define la anchura de la línea y afecta tanto a las líneas de los ejes como a las líneas creadas con lines.
<code>font =</code>	Controla la fuente (1, texto normal, 2 negrita, 3 cursiva, 4 = cursiva y negrita).

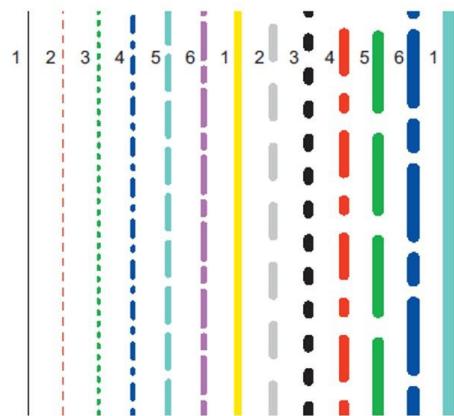
### Caracteres gráficos



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



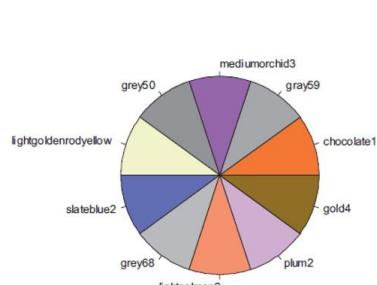
Tipos de Líneas



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



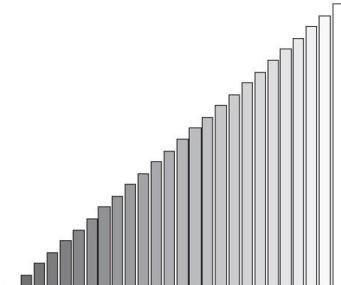
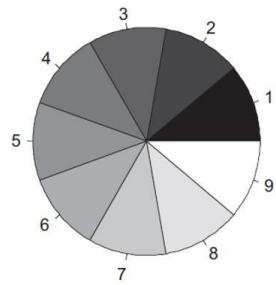
Paleta de colores



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Gama de grises



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Funciones Lattice

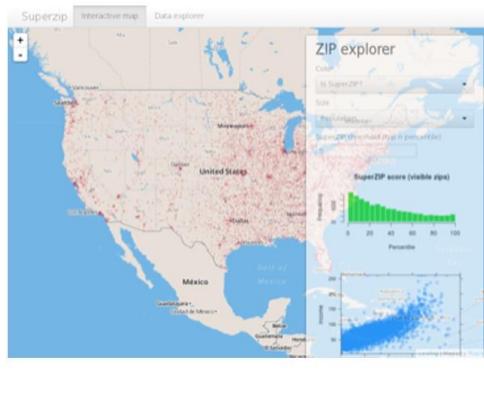
Xyplot	Gráfico bidimensional
Bwplot	Diagrama de cajas
Stripplot	Gráfico unidimensional para una variable continua y un factor
Dotplot	Gráfico de puntos
Histogram	Histograma
Densityplot	Gráfico de densidad
Barchart	Gráfico de barras
Piechart	Gráfico de sectores
Splom	Matrices gráficas

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Package ‘shiny’ November 2018  
Web Application Framework for R  
Version 1.2.0

```
install.packages("shiny")
runExample("01_hello")
http://127.0.0.1:5931/
```



```
if (!require("devtools"))
  install.packages("devtools")
  devtools::install_github("shiny", "rstudio")
```

<https://shiny.rstudio.com/gallery/>

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

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



<https://rmarkdown.rstudio.com/flexdashboard/examples.html>



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Web scraping y manipulación básica de datos

Técnica utilizada para extraer información de sitio web

<http://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000>

The screenshot shows a table titled "IBEX 35®" with columns: Nombre, Últ., % Dif., Max., Mín., Volumen, Efectivo (miles €), Fecha, and Hora. The data includes various stocks like ACCIONA, CERINER, ACS, ENCE, etc., with their respective values and time stamp.

Nombre	Últ.	% Dif.	Max.	Mín.	Volumen	Efectivo (miles €)	Fecha	Hora
ACCIONA	99.1400	0.45	99.9000	98.4000	109.198	10.815.16	27/03/2019	17:38
CERINER	8.7000	-0.75	8.8700	8.6720	840.643	8.743.27	27/03/2019	17:38
ACS	38.9000	2.03	39.8000	38.1000	679.819	28.715.00	27/03/2019	17:38
ENCE	102.9000	-0.72	103.5000	102.2000	71.453	27.247.75	27/03/2019	17:38
AMADEUS	68.9600	0.07	69.7400	68.1600	1.077.134	74.071.59	27/03/2019	17:38
ARCELORMIT	17.9600	0.10	18.1840	17.8100	640.744	9.702.59	27/03/2019	17:38
BASABADELL	0.8500	1.03	0.9020	0.8754	72.620.330	64.760.76	27/03/2019	17:38
BS SANTANDER	4.1520	1.57	4.2768	4.0762	56.497.301	23.552.46	27/03/2019	17:38
BNKIA	2.3400	0.95	2.4430	2.2500	9.694.853	23.588.05	27/03/2019	17:38
BRISTONINTER	8.0000	0.00	8.0700	7.9300	3.344.469	24.272.00	27/03/2019	17:38
CBU	5.1100	-0.52	5.2540	5.0920	32.475.404	167.513.88	27/03/2019	17:38
CAIXABANK	2.7950	1.82	2.8550	2.7380	19.182.437	53.502.96	27/03/2019	17:38
CELLnex	24.9600	-0.52	25.1400	24.7300	722.014	18.022.06	27/03/2019	17:38
CIE AUTOMOT	23.2200	1.22	23.9400	23.0000	182.149	4.244.31	27/03/2019	17:38

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Web scraping y manipulación básica de datos

Técnica utilizada para extraer información de sitio web

```
install.packages("rvest")
library(rvest)

descarga <- "http://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000"
dados <- read_html(descarga)
dados <- html_nodes(dados, "table")
length(dados)
```

RGui (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> descarga <- "http://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000"
> dados <- read_html(descarga)
> dados <- html_nodes(dados, "table")
> length(dados)
[1] 5
> dados
[1] <table class="Tb1Port" ...>
[2] <table cellpadding="0" cellspacing="0" align="center">...</table>
[3] <table cellpadding="5" cellspacing="2" class="TmBsqnq" style="width: 60%; align:center"><tr><td style="margin: 0 0 0px; vertical-align: middle;"><div><a href="#">...</a></div></td><td class="Cmpo">...</td></tr></table>
[4] <table class="Tb1Port" ...>
[5] <table class="Tb1Port" ...>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Web scraping y manipulación básica de datos

```
sapply(datos, class) # Verificar - Analizar la Dimensión
sapply(datos, function(x) dim(html_table(x, fill =
TRUE))) # Examinar la tabla de datos
indicador <- html_table(datos[[5]]) # Generar Tabla de
Indicadores a analizar
```

	Nombre	Ult. % Dif.	Max.	Min.	Volumen Efectivo (miles €)	Fecha	Hora	
1	ACCIONA	0,45	99,5000	98,4000	109.198	10.815,16	27/03/2019 Cierre	
2	ACERINBOX	0,7063	-0,75	8,8760	8,6720	848.643	7.428,27	27/03/2019 Cierre
3	ALCATEL RCS	0,59	2,28	35,6000	38,7500	679,453	26,781,25	27/03/2019 Cierre
4	AMADEUS	16,3500	1,00	140,0000	139,0000	17.453	10,717,75	27/03/2019 Cierre
5	AMADEUS	0,60	69,7400	68,1600	1.077,184	74.073,59	27/03/2019 Cierre	
6	ARCELORMITT.	0,7860	0,10	18,1840	17,8100	540.744	9.702,84	27/03/2019 Cierre
7	BA.SABADELL	0,8850	1,03	0,8200	0,8784	72.920.350	64.790,76	27/03/2019 Cierre
8	BA.SANTANDER	0,4200	4,9200	47,9500	47,9500	2.000.000	22,000,00	27/03/2019 Cierre
9	BANXIA	2,3460	0,95	2,4430	2,3050	9.984,853	23.589,08	27/03/2019 Cierre
10	BANKINTER	6,9480	0,77	6,9780	6,7580	3.537,085	24.225,90	27/03/2019 Cierre
11	BBVA	5,1160	-0,52	5,2840	5,0930	32.475,664	167.513,80	27/03/2019 Cierre
12	CADIZCOSTA	2,7500	0,20	2,7200	2,7800	19.522,457	18.522,457	27/03/2019 Cierre
13	CFIUSMEX	24,9600	-0,57	25,1400	24,7300	72.014	18.027,06	27/03/2019 Cierre
14	CIE AUTOMOT.	23,2200	1,22	23,6400	23,0000	182.149	4.244,31	27/03/2019 Cierre
15	ENAGAS	26,1000	-0,80	26,1300	25,8100	1.065,687	27.729,96	27/03/2019 Cierre
16	ENCE	4,7880	1,79	4,8660	4,6660	4.143,399	18.742,46	27/03/2019 Cierre
17	ENERGIA	23,2000	0,00	23,2000	23,2000	2.222,222	18.222,222	27/03/2019 Cierre
18	FERROVIAL	20,8100	-0,14	20,8700	20,5400	1.557,301	32.360,84	27/03/2019 Cierre
19	GRIFOLS CL.A	23,9200	-0,42	24,2800	23,8100	610.480	14.654,74	27/03/2019 Cierre
20	IAE	6,0380	1,21	6,0660	5,9660	2.273,363	13.704,10	27/03/2019 Cierre
21	IBERDROLA	7,9100	0,00	7,9100	7,9100	1.197,197	12.197,197	27/03/2019 Cierre
22	INDITEX	25,9000	-0,01	26,3000	25,8200	4.747,647	123.209,16	27/03/2019 Cierre
23	INORA A	9,6180	-0,67	9,8150	9,5850	513,304	4.976,74	27/03/2019 Cierre
24	INM.CENTRAL	9,1700	0,33	9,1850	9,0100	770.350	7.034,11	27/03/2019 Cierre
25	MOL	2,7000	0,00	2,7000	2,6900	4.000,000	18.000,00	27/03/2019 Cierre
26	MEDIASET	6,8740	-0,59	6,8250	6,5760	1.653,663	11.192,10	27/03/2019 Cierre
27	MELIA HOTELS	8,1600	0,87	8,2850	8,0850	576.059	4.724,61	27/03/2019 Cierre
28	MERLIN	11,4800	0,69	11,6800	11,5850	1.106,113	12.889,91	27/03/2019 Cierre
29	NATURGY	26,0500	0,44	25,0500	24,6200	817.333	20.362,51	27/03/2019 Cierre

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com

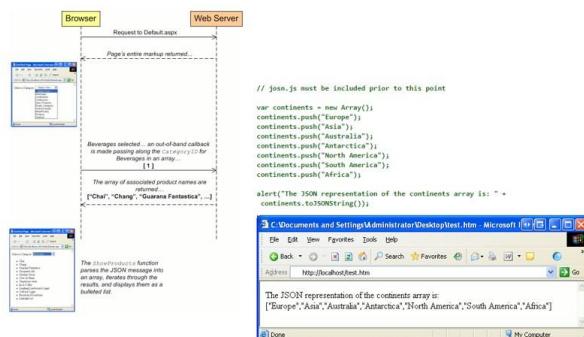


### Manipulación de texto

```
R Gui (64-bit) - [R Console]
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
> gsub ("o","d", c("oSWALDo", "Diaz"))
[1] "dSWALDd" "Diaz"
> grep ("W",c("oSWALDo", "Diaz"))# busca cadena de expresión regular
[1] 1
> paste("Hoy es ", date(), " y tengo clase de R", sep = "")
[1] "Hoy es Wed Mar 27 12:46:49 2019 y tengo clase de R"
> paste("!", 1:10, sep = "", collapse = "")
[1] "|_1, |_2, |_3, |_4, |_5, |_6, |_7, |_8, |_9, |_10"
> strsplit("oSWALDo Diaz", split = " ")
[1]
[1] "oSWALDo" "Diaz"

> strsplit(c("cadena de texto", "repositorio de datos"), split = " ")
[[1]]
[1] "cadena" "de"           "texto"
[[2]]
[1] "repositorio" "de"       "datos"
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



## [JSON]

JavaScript Object Notation

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



[http://servicios.ine.es/wstempus/js/ES/DATOS\\_TABLA/2852?nult=5&tip=AM](http://servicios.ine.es/wstempus/js/ES/DATOS_TABLA/2852?nult=5&tip=AM)

```
[{"COD": "DPOPI1", "Nombre": "Total Nacional. Total. Total habitantes. Personas.", "T3_Unidad": "Personas", "T3_Escala": "", "MetaData": [{"Id": 16473, "Variable": {"Id": 70, "Nombre": "Comunidades y Ciudades Aut\u00f3nomas", "Codigo": "10"}}, {"Id": 451, "Variable": {"Id": 18, "Nombre": "Sexo", "Codigo": "2"}, "Nombre": "Total", "Codigo": "0"}], "Id": 8677, "Variable": {"Id": 34, "Nombre": "Tama\u00f1o de los municipios", "Codigo": "3"}, "Nombre": "Total habitantes", "Codigo": "0"}, "Id": 20258, "Variable": {"Id": 3, "Nombre": "Tipo de dato", "Codigo": "5"}, "Nombre": "Persons", "Codigo": "0"}]
```

["Data": [{"Fecha": "2018-01-01T00:00:00+01:00", "T3\_TipoDato": "Definitivo", "T3\_Periodo": "A", "Anyo": 2018, "Valor": 4.672298E7}, {"Fecha": "2017-01-01T00:00:00+01:00", "T3\_TipoDato": "Definitivo", "T3\_Periodo": "A", "Anyo": 2017, "Valor": 4.6572132E7}, {"Fecha": "2016-01-01T00:00:00+01:00", "T3\_TipoDato": "Definitivo", "T3\_Periodo": "A", "Anyo": 2016, "Valor": 4.6557008E7}, {"Fecha": "2015-01-01T00:00:00+01:00", "T3\_TipoDato": "Definitivo", "T3\_Periodo": "A", "Anyo": 2015, "Valor": 4.6624382E7}]}

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



```
install.packages("rjson")
library(rjson)
```

R GUI (64-bit) - [R Console]

Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

```
> datos <- readLines("http://servicios.ine.es/wstempus/js/ES/DATOS_TABLA/2852?nult=5&tip=AM")
> datos <- paste(datos, collapse = " ") # separamos las filas
> datos <- fromJSON(datos)
> length(datos) # Validar cuantas filas
[1] 159
> class(datos) # Instanciar datos a objeto
[1] "list"
> datos[[89]]$Data[[5]]$Valor# Localizar un valor específico
[1] 166325
> datos[[89]]$Nombre
[1] "Lugo. Hombres. Total habitantes. Personas. "
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



XML eXtensible Markup Language ("Lenguaje de Marcas Extensibles - Etiquetas").

[http://api.worldbank.org/countries/all/indicators/NY.GDP.MKTP.CD?](http://api.worldbank.org/countries/all/indicators/NY.GDP.MKTP.CD)

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<wb:data xmlns:wb="http://www.worldbank.org" page="1" pages="6" per_page="50" total="264">
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="1A">Arab World</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="53">Caribbean small states</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="88">Central Europe and the Baltics</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="V2">Early-demographic dividend</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="Z4">East Asia & Pacific (excluding high income)</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="4E">East Asia & Pacific (high income)</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="T4">East Asia & Pacific (low income)</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="Z2">Europe and Central Asia</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="T5">Europe, Central and Eastern Europe</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="EU">European Union</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="F1">Fragile and conflict affected situations</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XE">Heavily indebted poor countries</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XD">High income</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XE2">IBRD only</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="Z7">IDA only</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XI">IDA total</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XO">IDA total</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



XML

```
install.packages("xml2")
library(xml2)
```

```
datos <- read_xml("http://api.worldbank.org/countries/all/indicators/NY.GDP.MKTP.CD?date=2018")
```

```
R Gui (64-bit) - [R Console]
```

```
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
```

```
library(xml2)
datos <- read_xml("http://api.worldbank.org/countries/all/indicators/NY.GDP.MKTP.CD?date=2018")
datos
```



```
[xml_document]
<?xml version="1.0" encoding="UTF-8"?>
<wb:data xmlns:wb="http://www.worldbank.org" page="1" pages="6" per_page="50" total="264">
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="1A">Arab World</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="53">Caribbean small states</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="88">Central Europe and the Baltics</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="V2">Early-demographic dividend</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="Z4">East Asia & Pacific (excluding high income)</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="4E">East Asia & Pacific (high income)</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="T4">East Asia & Pacific (low income)</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="Z2">Europe and Central Asia</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="T5">Europe, Central and Eastern Europe</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
  <wb:country id="EU">European Union</wb:country>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="F1">Fragile and conflict affected situations</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XE">Heavily indebted poor countries</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XD">High income</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XE2">IBRD only</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="Z7">IDA only</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XI">IDA total</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
  <wb:indicator id="XO">IDA total</wb:indicator>
  <wb:date>2018</wb:date>
  <wb:value/>
  <wb:decimal>0</wb:decimal>
</wb:indicator>
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Exportando datos CSV

#### Paquete CSV

```
install.packages("csv",dependencies=TRUE)
library(csv)
```

- Ingresas datos a analizar con la siguiente estructura: Factor <-> variables (Edad, Estatura, Peso, Nacionalidad, Genero, Escuela de Procedencia)
- Agrupa los factores en un conjunto y nombre las variables.
- Convertir toda la información en una base y posteriormente darle formato de tabla.

```
datos<-data.frame(X)
write.table(datos,"Estadistica Grupo 1.txt")
```

- Exporta la tabla a un archivo excel.

```
write.csv(datos,"Estadistica Grupo 1.csv")
```

```
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.

> install.packages("csv",dependencies=TRUE)
--- Please select a CRAN mirror for use in this session ---
probando la URL 'https://cran.itam.mx/bin/windows/contrib/3.5/csv_0.5.2.zip'
Content type 'application/zip' length 22937 bytes (22 KB)
downloaded 22 KB

package 'csv' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/edgaroswaldo.diaz/AppData/Local/Temp/RtmpQ9TxDE/downloaded_packages
> library(csv)
> Edad <- c(23,30,30,35,40)
> Estatura <- c(1.64,1.70,1.75,1.80,1.85)
> Peso <- c(50,70,85,45,80)
> Nacionalidad<-c("Mexicano","Mexicano","Extranjero","Extranjero")
> Genero <- c("Mujer","Hombre","Mujer","Hombre")
> X <-cbind(Edad,Estatura,Peso,Nacionalidad,Genero)
> row.names(X) <- c("Ana","Alberto","Anyelo","Fulanita","Sultanito")
> datos<-data.frame(X)
> datos<-table(datos,"Estadistica Grupo 1.txt")
> write.table(datos,"Estadistica Grupo 1.csv")
> dir()
[1] "Estadistica Grupo 1.csv"  "Estadistica Grupo 1.txt"  "open.exe"      "R.dll"
[5] "R.exe"                   "Rblas.dll"        "Rcmd.exe"     "Rfe.exe"
[9] "Rgraphpack.dll"         "Rgui.exe"        "Riconv.dll"   "Rlapack.dll"
[13] "Rscript.exe"            "Rterm.exe"       "RtermReg.exe"

> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldo.diaz2375@gmail.com



### Importando datos

```
R Gui (64-bit)
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
R Gui (64-bit)
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda
R Console
> read.csv("Estadistica Grupo 1.csv")
  X Edad Estatura Peso Nacionalidad Genero
1 Ana 23 1.64 50 Mexicano Mujer
2 Alberto 30 1.70 70 Mexicano Hombre
3 Anyelo 30 1.80 85 Mexicano Hombre
4 Fulanita 35 1.85 45 Extranjero Mujer
5 Sultanito 40 1.80 80 Extranjero Hombre
> Datos <- read.csv("Estadistica Grupo 1.csv")
> View(Datos)
> |
```

```
Data: Datos
  X | Edad | Estatura | Peso | Nacionalidad | Genero
1 Ana | 23 | 1.64 | 50 | Mexicano | Mujer
2 Alberto | 30 | 1.70 | 70 | Mexicano | Hombre
3 Anyelo | 30 | 1.80 | 85 | Mexicano | Hombre
4 Fulanita | 35 | 1.85 | 45 | Extranjero | Mujer
5 Sultanito | 40 | 1.80 | 80 | Extranjero | Hombre
```

(Big Data) | Análisis de Datos con R | edgaroswaldo.diaz2375@gmail.com



### Paquete XLSX

```
install.packages("xlsx",dependencies = TRUE)  
library(xlsx)
```

```
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda  
File Edit View More Packages Window Help  
> install.packages("xlsx",dependencies = TRUE)  
also installing the dependencies 'rJava', 'xlsxjars'  
  
probando la URL 'https://cran.itam.mx/bin/windows/contrib/3.5/rJava_0.9-10.zip'  
Content type 'application/zip' length 825992 bytes (806 KB)  
downloaded 806 KB  
  
probando la URL 'https://cran.itam.mx/bin/windows/contrib/3.5/xlsxjars_0.6.1.zip'  
Content type 'application/zip' length 9485464 bytes (9.0 MB)  
downloaded 9.0 MB  
  
probando la URL 'https://cran.itam.mx/bin/windows/contrib/3.5/xlsx_0.6.1.zip'  
Content type 'application/zip' length 458979 bytes (448 KB)  
downloaded 448 KB  
  
Package 'rJava' successfully unpacked and MD5 sums checked  
Package 'xlsxjars' successfully unpacked and MD5 sums checked  
Package 'xlsx' successfully unpacked and MD5 sums checked  
  
The downloaded binary packages are in  
C:\Users\oswaldo.diaz\AppData\Local\Temp\RtmpQ9TxDE\downloaded_packages  
> library(xlsx)  
> |
```

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Paquete XLSX

Parámetro	Descripción
SheetName	Nombre de la hoja del libro en que se introducirán los datos.
Append	Por defecto toma el valor FALSE, provocando que la función cree un libro con el nombre indicado por filename eliminándolo si ya existiese. Si es TRUE el libro debe existir y al mismo se añadirá una hoja con los nuevos datos.
Col.names y Row.names	Determinan si se escribirán en la hoja los nombres asociados a las columnas y las filas del data frame. Por defecto ambos parámetros toman el valor TRUE.
Función	Descripción
dataFormat	Formato.
alignment	Alineación.
border	bordes de celda.
font	Fuente del texto.
row	Contenido de filas.
colIndex	Especifica el índice de las columnas.
value	Variable de longitud.
richTextString	Indica si el contenido debe insertarse como texto enriquecido.
showNA	Valor lógico. Si es VERDADERO la celda contendrá el valor "NA". Si es Falso, se saltarán.
keepFormulas	Un valor lógico. Si es VERDADERA las fórmulas se devuelven como caracteres en lugar de ser evaluadas explícitamente.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Paquete XLSX

Función	Descripción
sheetName	Nombre de la hoja que se va a modificar.
createSheet	Función para crear una hoja.
removeSheet	Función para eliminar una hoja.
Función	Descripción
xlsx.addTitle	Función para agregar título y subtítulo.
CellStyle	Función para crear estilos de celdas.
TABLE_ROWNAMES_STYLE	Función para el formato de filas.
TABLE_COLNAMES_STYLE	Función para el formato de columnas.
TITLE_STYLE	Función para el formato de título.
SUB_TITLE_STYLE	Función para el formato de subtítulos.

Dentro de la estructura de las funciones se encuentran comandos con diversos argumentos. Tres de ellos son: Font, Alignment y Border.

- **Font** Color (color de la fuente), HeightInPoints (tamaño de letra), IsBold/italic (indica si la fuente debe estar en negrita o cursiva), Underline (grosor del subrayado) y name (fuente a utilizar).
- **Alignment** WrapText (indica si la celda debe ajustarse para mostrar el texto completo), alineación horizontal ("ALIGN\_CENTER", "ALIGN\_JUSTIFY", "ALIGN\_LEFT", "ALIGN\_RIGHT"), alineación vertical ("VERTICAL\_BOTTOM", "VERTICAL\_CENTER", "VERTICAL\_JUSTIFY", "VERTICAL\_TOP") y rotation (especifica los grados que se desea girar el texto en la celda).
- **Border** Color (es el color del borde), position ("bottom", "left", "top", "right") y pen (estilo de la línea del borde, por ejemplo:"border\_dash\_dot", "border\_dash\_dot\_dot", "border\_slanted\_dash\_dot", etc.).

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



### Paquete XLSX

Función	Descripción
addDataFrame	Función para agregar una tabla. StartRow y startColumn son un valor numérico que indican la fila y columna inicial. ColnameStyle y rownameStyle personalizan los nombres de encabezado y fila de la tabla.
setColumnWidth	Función para cambiar el ancho de una columna. ColIndex es un vector numérico que indica las columnas que se desea cambiar el grosor, por su parte ColWidth cambia el ancho de la columna.
png	Función para crear una gráfica png.
addPicture	Función para añadir gráficas png.
file.remove	Función para eliminar gráficas.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



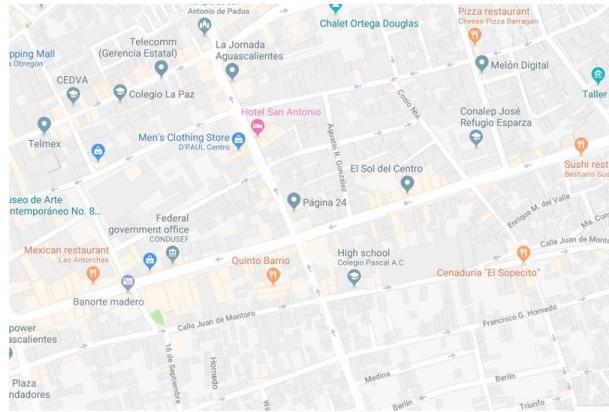
ggmap

```
install.packages("ggmap")
library(ggmap)
```

```
localizar <- geocode('Francisco I. Madero, Zona
Centro, Aguascalientes', source = "google")
mapa.localizar <- get_map(location =
as.numeric(localizar),
color = "color",
maptype = "roadmap",
scale = 2,
zoom = 16)
```

```
ggmap(mapa.localizar) + geom_point(aes(x =
lon, y = lat), data = localizar, colour = 'red', size
= 4)
```

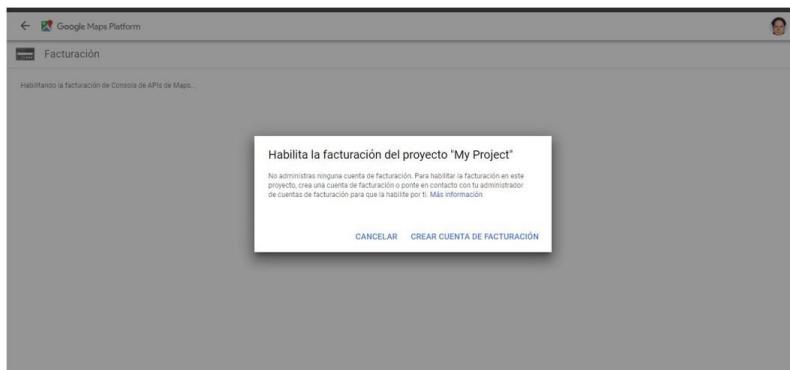
@21.8828997,-102.291656,17z



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



ggmap



(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



Pagas todos los meses a través de un cargo automático cuando vence el pago.

Método de pago ⓘ

Número de tarjeta	MM	AA	CVC
#	/		
Se requiere el número de tarjeta	Se requiere el mes	Se requiere el año	Se requiere el CVC

Titular de la tarjeta  
Edgar Oswaldo Díaz

La dirección de la tarjeta de crédito o de débito es la misma que figura arriba.

[INICIAR VERSIÓN DE PRUEBA GRATUITA](#)

[Política de privacidad](#) | [Preguntas frecuentes](#)

**Acceso a todos los productos de Cloud Platform**  
Consigue todo lo que necesitas para desarrollar y ejecutar tus aplicaciones, sitios web y servicios, incluidos Firebase y la API de Google Maps.

**Crédito de 300 \$ gratuito**  
Regístrate y consigue 300 \$ para gastarlos en Google Cloud Platform durante los próximos 12 meses.

**Sin cargos automáticos cuando finaliza el periodo de prueba gratuito**  
Solo te pedimos los datos de la tarjeta de crédito para comprobar que no eres un robot. No se te cobrará nada a menos que actualices la cuenta manualmente a la versión de pago.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



#### SKU: Geocodificación

Un SKU de **geocodificación** se cobra por las solicitudes a la [API de geocodificación](#) o al [servicio de geocodificación de la API de JavaScript de Google Maps](#).

GAMA DE VOLUMEN MENSUAL (Precio por SOLICITUD)		
0–100,000	100,001–500,000	500,000+
0.005 USD por cada uno (5.00 USD por 1000)	0.004 USD por cada uno (4.00 USD por 1000)	Contacte a <a href="#">Ventas</a> para precios por volumen

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com



- D. M. Bates and D. G. Watts (1988), Nonlinear Regression Analysis and Its Applications. John Wiley & Sons, New York.
- Richard A. Becker, John M. Chambers and Allan R. Wilks (1988), The New S Language. Chapman & Hall, New York. This book is often called the “Blue Book”.
- John M. Chambers and Trevor J. Hastie eds. (1992), Statistical Models in S. Chapman & Hall, New York. This is also called the “White Book”.
- John M. Chambers (1998) Programming with Data. Springer, New York. This is also called the “Green Book”.
- A. C. Davison and D. V. Hinkley (1997), Bootstrap Methods and Their Applications, Cambridge University Press.
- Annette J. Dobson (1990), An Introduction to Generalized Linear Models, Chapman and Hall, London.
- Peter McCullagh and John A. Nelder (1989), Generalized Linear Models. Second edition, Chapman and Hall, London.
- John A. Rice (1995), Mathematical Statistics and Data Analysis. Second edition. Duxbury Press, Belmont, CA.
- S. D. Silvey (1970), Statistical Inference. Penguin, London.

(Big Data) | Análisis de Datos con R | edgaroswaldodiaz2375@gmail.com