

Bioestadística con

Madrid, Noviembre y Diciembre de 2019

Jesús Herranz Valera
Bioestadístico de GEICAM

Programa

- | | |
|---|---|
| 1. Introducción a R | 13. Regresión Logística I |
| 2. Variables y objetos de R | 14. Regresión Logística II |
| 3. Manejo de datos | 15. Variables de confusión e Interacciones |
| 4. Gráficos en R | 16. Construcción de un modelo de regresión |
| 5. Estadística descriptiva | |
| 6. Programación y funciones en R | 17. Análisis de Supervivencia |
| | 18. Regresión de Cox I |
| 7. Tablas de Contingencia | 19. Regresión de Cox II |
| 8. Inferencia básica | |
| 9. Análisis de la varianza | 20. Modelos predictivos. Curvas ROC |
| 10. Análisis de Correlación | 21. Análisis Medidas repetidas |
| | 22. Comparaciones Múltiples |
| 11. Regresión lineal simple | 23 Análisis de Componentes Principales |
| 12. Regresión lineal múltiple | 24 Análisis Cluster |

Contenido del material didáctico

- **Copiar la carpeta “Bioestadistica con R” a “C:”**
- **Datos**
 - Ficheros de datos de los ejemplos y ejercicios del curso
- **R Scripts**
 - R scripts usados como ejemplos en el curso
- **Ejercicios resueltos**
 - R scripts con los ejercicios resueltos
- **Otras carpetas**
 - **Documentos** interesantes relacionados con R
 - **Resumen de funciones de R**
 - **Software Tinn R**

Bibliografía

- W. Venables. *Modern Applied Statistics with S*. Springer, 2002
- **B. Everitt & S. Rabe-Hesketh.** *Analysing Medical Data using S-PLUS*. Springer, 2001
- J. Pinheiro & D. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer, 2000
- B. Everitt. *An R and S-PLUS Companion to Multivariate Analysis*. Springer, 2007
- N. Matloff. *The Art of R Programming*. No Starch Press, 2001
- D. Montgomery & E. Peck. *Introduction to Linear Regression Analysis*. Wiley, 1992
- **D. Hosmer & S. Lemeshow.** *Applied Logistic Regression*. Wiley, 2000
- **D. Hosmer & S. Lemeshow.** *Applied Survival Analysis*. Wiley, 2008
- D. Kleinbaum & M. Klein. *Logistic Regression*. Wiley, 2010
- D. Kleinbaum & M. Klein. *Survival Analysis*. Wiley, 2012
- E. Sánchez-Cantalejo. *Regresión Logística en Salud Pública*. EASP, 2000
- F. Harrell. *Regression Modeling Strategies*. Springer, 2001

Estadística Aplicada a la Investigación Biomédica con R

Material complementario

- ✓ Descripción de los ficheros de datos

Fichero de datos: bajo peso al nacer

- Objetivo del estudio: se desea conocer factores que están asociados al bajo peso de los recién nacidos, presentes en mujeres que han dado a luz

Nombre	Descripción	Categorías
id	Identificador	
bajo_pes	Bajo peso al nacer	0 = paso normal (≥ 2500gr.) 1 = bajo peso (< 2500 gr.)
edad	Edad	
peso	Peso	
raza	Raza	1 = blanca, 2 = negra, 3 = otras
fumador	Fumador	1 = si, 0 = no
part_pre	Partos prematuros	0 = no, 1 = 1 parto, 2 = 2 partos,
hta	Hipertensión	1 = si, 0 = no
irr_urin	Irritabilidad urinaria	1 = si, 0 = no
visi_med	Número de visitas al médico	

- Regresión logística. Estadística descriptiva, inferencia básica con variables continuas, tablas de contingencias

Fichero de datos: umaru

- Objetivo del estudio: comparar 2 programas de tratamiento de diferente duración (corto y largo) diseñados para reducir el abuso de drogas

Nombre	Descripción	Categorías
ID	Identificador	
AGE	Edad	
BECK	Score Beck de depresión	0 – 54.000
IVHX	Historia de consumo de drogas	1 = nunca, 2 = previa, 3 = reciente
NDRUGTX	Número previo de drogas	0 – 40
RACE	Raza	0 = blanca, 1 = otras
TREAT	Tratamiento	0 = corto, 1 = largo (duración del tratamiento)
SITE	Lugar donde se realizó el tratamiento	0 = sitio A, 1 = sitio B
DFREE	Retorno al consumo de drogas después 12 meses	0 = libre de consumo, 1 = recayó en el consumo

- Regresión logística. Estadística descriptiva, inferencia básica con variables continuas, tablas de contingencias

Fichero de datos: cystic fibrosis

- Objetivo del estudio: examinar en pacientes con fibrosis quística la relación de mala nutrición con variables relacionadas con el tamaño corporal y funciones pulmonares

Nombre	Descripción	Categorías
sub	Identificador	
age	Edad	
sex	Sexo	1 = mujer, 0 = hombre
height	Altura	
weight	Peso	
bmp	Índice de masa corporal	
fev	Volumen respiratorio	
rv	Medida del volumen pulmonar	
frc	Capacidad residual funcional	
tlc	Capacidad pulmonar total	
pemax	Medida de malnutrición	

- Regresión lineal

Fichero de datos: whas500 (1)

- Objetivo del estudio: explorar factores asociados a la supervivencia, después de sufrir un infarto de miocardio (MI)

Nombre	Descripción	Categorías
id	Identificador	
age	Edad	
gender	Sexo	0 = hombre, 1 = mujer
hr	Ritmo cardiaco	
sysbp	Presión sistólica	
diasbp	Presión diastólica	
bmi	Índice de masa corporal	
cvd	Historia enferm. Cardiovascular	0 = no, 1 = si
afb	Fibrilación arterial	0 = no, 1 = si

- Análisis de supervivencia

Fichero de datos: whas500 (2)

Nombre	Descripción	Categorías

sho	Shock cardiaco	0 = no, 1 = si
chf	Complicaciones cardiacas	0 = no, 1 = si
av3	Bloqueo cardiaco	0 = no, 1 = si
miord	Orden del infarto	0 = primero, 1 = recurrente
mitype	Tipo del infarto	0 = no Q-wave, 1 = Q-wave
lenfol	Tiempo de Seguimiento	Diferencia entre la fecha de admisión en el hospital y la última fecha de seguimiento
fstat	Estado vital	0 = vivo, 1 = muerto

- Análisis de supervivencia

Fichero de datos: actg320 (AIDS clinical trial gr.)

- Objetivo del estudio: examinar la efectividad de un nuevo tratamiento de 3 fármacos en la supervivencia en enfermos de HIV

Nombre	Descripción	Categorías
id	Identificador	
time	Tiempo hasta aparece el SIDA o muerte	
censor	Evento indicador	1 = SIDA o muerte, 0 = observación censurada
tx	Tratamiento	1 = nuevo tratamiento 3 fármacos; 0 = trat. 2 fármacos
sex	Sexo	1 = hombre, 2 = mujer
ivdrug	Uso de IV drogas	1 = nunca, 2 = actualmente, 3 = previamente
hemophil	Hemofilia	1 = si, 0 = no
cd4	Conteo de CD4 baseline	
priorzdv	Meses de uso de ZDV	
age	Edad	

- Análisis de supervivencia

Fichero de datos: evolución cirugía

- Objetivo del estudio: analizar la evolución de unos parámetros clínicos a lo largo del tiempo, después de una intervención quirúrgica, en 4 tipos de operación distintos

Nombre	Descripción	Categorías
id	Identificador	
tiempo	Momento del tiempo	1 = momento 1º: medida basal antes de la intervención 2 = momento 2º durante la intervención 3 = momento 3º durante la intervención 4 = momento 4º: medida al finalizar la intervención
grupo	Tipo de operación	4 operaciones distintas codificadas del 1-4
sat	Sat (parámetro)	
pvc	Pvc (parámetro)	

- Análisis medidas repetidas

Fichero de datos: Virco

- **Objetivo del estudio:** analizar la relación de la **secuencia de la Proteasa** con la **resistencia a inhibidores** de la proteasa en 976 muestras virales en enfermos de HIV

Nombre	Descripción	Categorías / Observaciones
P1-P99	89 variables que contienen la información genética XX=posición del aminoácido dentro de la región de la Proteasa del genoma viral	Variables Binarias: 0 = no mutado (AA, más común en la población) 1 = mutado
sens.NFV	Sensibilidad a NFV (Nelfinavir)	0 si es menos sensible a NFV que a IDV (Indinavir) 1 si es más sensible a NFV que a IDV

- Problema de **Clasificación**, usando la variable respuesta binaria “**sens.NFV**”
- Se ha creado otro fichero , “Virco_data_Missings.csv”, donde se han asignado aleatoriamente missings a algunos valores de las variables predictoras
- Ejemplo extraído de “*Applied Statistical Genetics with R*”. Andreas S. Foulkes

Fichero de datos: ALL

- **Objetivo del estudio:** estudiar qué **genes** están relacionados con **2 tipos de tumores** en pacientes de leucemia (ALL, acute lymphoblastic leukemia)

Nombre	Descripción	Categorías / Observaciones
X1 – X12625	12625 Variables con datos de expresión génica	Variables continuas
mol.biol	Tipo de mutación	BCR/ABL = tumores con esta mutación NEG = tumores sin anormalidad citogenética

- Problema de **Clasificación**, usando la variable respuesta binaria “**mol.biol**”
- Está disponible en Bioconductor
- Existe otro fichero llamado “**ALLSubset**” con solo 1000 variables genéticas

Fichero de datos: Prostate

- **Objetivo del estudio:** estudiar la relación entre el nivel de **PSA** (prostate specific antigen) con algunas **variables clínicas**, en pacientes que han recibido una prostatectomía radical

Nombre	Descripción	Categorías / Observaciones
Ipsa	PSA	Variable continua. Escala logarítmica
Icavol	Volumen del cáncer	Escala logarítmica
Iweighth	Peso de la próstata	Escala logarítmica
age	Edad	
Ipph	Cantidad hiperplasia benigna	Escala logarítmica
svi	Invasión vesícula	0 = No, 1 = Si
Icp	Penetración capsular	Escala logarítmica
gleason	Gleason Score	Ordinal con valores entre 6 y 9
pgg45	Porcentaje de Gleason 4 o 5	Continua, entre 0 y 100

- Problema de **Regresión**, usando la variable respuesta continua “**Ipsa**”
- Ejemplo extraído de “*The Elements of Statistical Learning*”. *Trevor Hastie*

Fichero de datos: NCI Microarray

- **Objetivo del estudio:** encontrar **grupos** similares en la **expresión génica** de **64 tumores**

Nombre	Descripción	Categorías / Observaciones
	6830 Genes	Datos de Expresión
Type Tumors	Tipo de tumor	BREAST, CNS, COLON, MELANOMA, RENAL, PROSTATE,

- Problema de **Clustering**, usando todas las variables menos el Tipo de Tumor, que se utilizará solamente como una etiqueta para tratar de entender los grupos encontrados
- Los datos genéticos están en un fichero donde las filas son las variables y las columnas indican los datos de cada tumor (estructura común en datos genéticos). Hay que trasponer el fichero
- El Tipo de Tumor está en un fichero aparte
- Ejemplo extraído de “*The Elements of Statistical Learning*”. *Trevor Hastie*

Estadística Aplicada a la Investigación Biomédica con R

1 Introducción a R

- ✓ **Introducción a R**
- ✓ **Instalación de R**
- ✓ **Paquetes de R**
- ✓ **Primeras nociones. El entorno de R**

Software R

- **R es un lenguaje y un entorno de programación para análisis de datos y gráficos**
 - Lenguaje de programación simple e intuitivo. Orientado a objetos
 - Conjunto amplio de funciones de análisis de datos
 - Funciones muy flexibles para realizar gráficos
 - Almacenamiento y manejo de datos
 - Se ejecuta en una línea de comandos. **Consola de R**
- R es una implementación “**open-source**” de S, base del software S-PLUS
 - Bastante compatibles
- R se distribuye bajo **Windows, Linux y MacOS**
 - Bastante compatibles

Ventajas e inconvenientes de R

- **Ventajas**
 - Combina **funciones estadísticas y programación**
 - Realización de **gráficos** de alta calidad
 - R es **gratis**: R es “portable”
 - R es muy **potente** y **flexible** para incorporar nuevas técnicas estadísticas
 - Se pueden **construir funciones** propias fácilmente
 - Comunidad de R es muy **dinámica**. Muy implicada en el desarrollo de R
- **Inconvenientes**
 - No tiene un **entorno** “amigable”
 - **Aprendizaje** más costoso
 - Cambio de **versiones** de R y mantenimiento de los paquetes. R no tiene garantía
 - El funcionamiento de cada **paquete** es diferente

Instalación de R (www.r-project.org)

The screenshot shows the homepage of the R Project for Statistical Computing. The URL in the browser bar is https://www.r-project.org/. The page features a large R logo on the left, followed by a navigation menu with links like Home, Download (which is highlighted with a red box), CRAN, R Project (with sub-links: About R, Contributors, What's New?, Mailing Lists, Bug Tracking, Conferences, Search), R Foundation (with sub-links: Foundation, Board, Members, Donors, Donate), Documentation (which is also highlighted with a red box), and Links (with sub-links: Bioconductor, Related Projects). The main content area is titled "The R Project for Statistical Computing" and includes sections for "Getting Started" (describing R as a free software environment for statistical computing and graphics, mentioning CRAN mirrors for download) and "News" (listing recent releases like R version 3.2.2, The R Journal Volume 7/1, R version 3.1.3, useR! 2015, and useR! 2014). A sidebar on the right contains a search bar and other site navigation.

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

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and Mac OS. To [download R](#), please choose your preferred [CRAN](#) mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- [R version 3.2.2 \(Fire Safety\)](#) has been released on 2015-08-14.
- [The R Journal Volume 7/1](#) is available.
- [R version 3.1.3 \(Smooth Sidewalk\)](#) has been released on 2015-03-09.
- [useR! 2015](#), will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.
- [useR! 2014](#), took place at the University of California, Los Angeles, USA June 30 - July 3, 2014.

Instalación de R. CRAN (<https://cran.r-project.org>)

The screenshot shows a web browser displaying the CRAN homepage at <https://cran.r-project.org/>. The page title is "The Comprehensive R Archive Network". The left sidebar contains links for CRAN Mirrors, What's new?, Task Views, Search, About R, R Homepage, The R Journal, Software, R Sources, R Binaries, Packages (which is highlighted with a red box), and Other. The main content area has a red box around the "Download and Install R" section. It contains instructions for precompiled binary distributions and links for Download R for Linux, Mac OS X, and Windows. Below this is a section for Source Code for all Platforms, which is intended for Windows and Mac users. The "Questions About R" section at the bottom provides answers to frequently asked questions.

The Comprehensive R Archive Network

Download and Install R

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

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

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

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-08-14, Fire Safety) [R-3.2.2.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

Instalación de R

The screenshot shows a web browser window with the URL <https://cran.r-project.org/> in the address bar. The title of the page is "R-3.2.2 for Windows (32/64 bit)". On the left, there is a large R logo and a sidebar with links for CRAN Mirrors, What's new?, Task Views, Search, About R, R Homepage, The R Journal, Software, R Sources, R Binaries, Packages, Other, Documentation, Manuals, FAQs, and Contributed. The main content area has a heading "Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)" which is highlighted with a red box. Below it are links for "Installation and other instructions" and "New features in this version". A note below the download link says: "If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the [md5sum](#) of the .exe to the [true fingerprint](#). You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available." There is also a "Frequently asked questions" section with links for "How do I install R when using Windows Vista?", "How do I update packages in my previous version of R?", and "Should I run 32-bit or 64-bit R?". Another section, "Other builds", lists "r-patched snapshot build", "r-devel snapshot build", and "Previous releases". At the bottom, it says: "Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.htm".

Last change: 2015-08-14, by Duncan Murdoch

Documentación de R

- “**Documentation**” (menú de la pantalla principal)
 - “**Manuals**”. Manuales oficiales de R como software y lenguaje de programación
 - “**The R Journal**”. 2 revistas anuales
 - “**Books**”. Libros relacionados con R
 - “**Other**”. Páginas web relacionadas con R
 - “Journal of Statistical Software”
- “**CRAN**”
 - “**Contributed**”. Documentos y libros de acceso libre de introducción a R y con contenidos estadísticos

Instalación de paquetes con R

- La instalación básica de R incorpora las técnicas estadísticas más simples
- Las técnicas más especializadas se instalan con **paquetes** adicionales
- Se encuentran disponibles en **CRAN** (Comprehensive R Archive Network)
- **Instalar paquetes**: opción de menú “Packages”
 - “Install Packages”
 - “Update Packages”, busca una nueva versión del paquete
 - “Install Packages from local ZIP file”, previamente se ha descargado desde CRAN
- También se puede instalar con: **install.packages("ggplot2")**
- R no tiene acceso a las librerías instaladas. Hay que usar la función **library()**

library()	Carga una librería en R
search()	Muestra la lista de librerías

Instalación de paquetes con R

The screenshot shows a web browser displaying the CRAN website at <https://cran.r-project.org/>. The page title is "Contributed Packages". On the left, there's a sidebar with links like "CRAN", "Mirrors", "What's new?", "Task Views", "Search", "About R", "R Homepage", "The R Journal", "Software", "R Sources", "R Binaries", "Packages" (which is highlighted with a red box), and "Other". Below that is a section for "Documentation" with links to "Manuals", "FAQs", and "Contributed". The main content area starts with "Available Packages" and a note that there are 7096 available packages. It includes two redboxed links: "Table of available packages, sorted by date of publication" and "Table of available packages, sorted by name". Below this is a "Installation of Packages" section with instructions and a link to the manual. The next section is "CRAN Task Views", followed by "Package Check Results", "Writing Your Own Packages", and finally "Repository Policies". Each of these sections has a link to their respective manuals.

Contributed Packages

Available Packages

Currently, the CRAN package repository features 7096 available packages.

[Table of available packages, sorted by date of publication](#)

[Table of available packages, sorted by name](#)

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual [R Installation and Administration](#) (also contained in the R base sources) explains the process in detail.

[CRAN Task Views](#) allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 33 views are available.

Package Check Results

All packages are tested regularly on machines running [Debian GNU/Linux](#), [Fedora](#) and Solaris. Packages are also checked under OS X and Windows, but typically only on the day the package appears on CRAN.

The results are summarized in the [check summary](#) (some [timings](#) are also available). Additional details for Windows checking and building can be found in the [Windows check summary](#).

Writing Your Own Packages

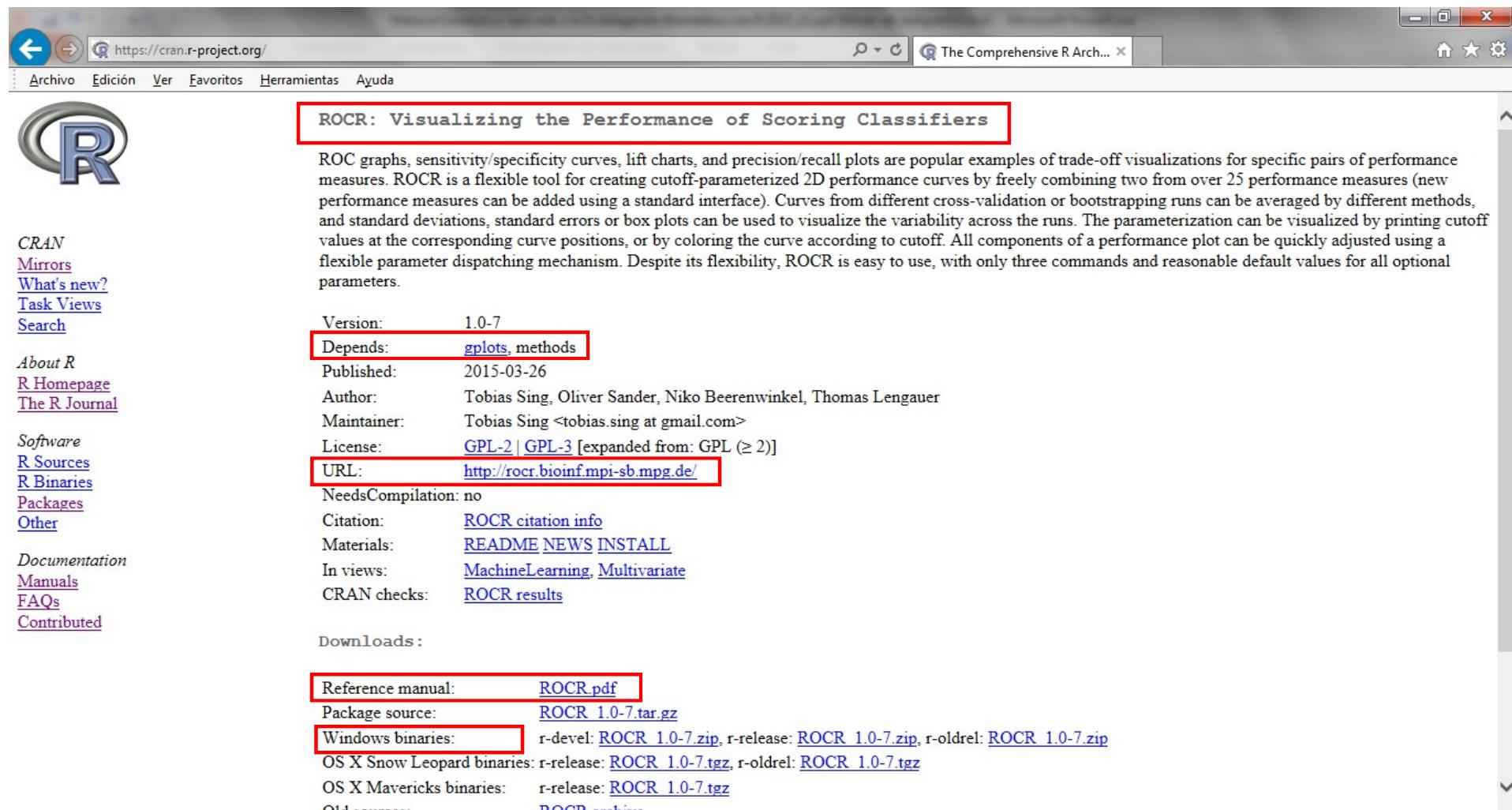
The manual [Writing R Extensions](#) (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Repository Policies

The manual [CRAN Repository Policy \[PDF\]](#) describes the policies in place for the CRAN package repository.

- Se muestra una lista de todos los paquetes disponibles de R

Instalación de paquetes con R



The screenshot shows a web browser displaying the CRAN package page for 'ROCR'. The page title is 'ROCR: Visualizing the Performance of Scoring Classifiers'. The page content includes a brief description of the package, version information (1.0-7), dependencies (gplots, methods), publication details (published 2015-03-26, authors Tobias Sing, Oliver Sander, Niko Beerenwinkel, Thomas Lengauer), maintainer information (Tobias Sing <tobias.sing at gmail.com>), license (GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]), URL (http://rocr.bioinf.mpi-sb.mpg.de/), citation information, and various download links for different operating systems.

ROCR: Visualizing the Performance of Scoring Classifiers

ROC graphs, sensitivity/specificity curves, lift charts, and precision/recall plots are popular examples of trade-off visualizations for specific pairs of performance measures. ROCR is a flexible tool for creating cutoff-parameterized 2D performance curves by freely combining two from over 25 performance measures (new performance measures can be added using a standard interface). Curves from different cross-validation or bootstrapping runs can be averaged by different methods, and standard deviations, standard errors or box plots can be used to visualize the variability across the runs. The parameterization can be visualized by printing cutoff values at the corresponding curve positions, or by coloring the curve according to cutoff. All components of a performance plot can be quickly adjusted using a flexible parameter dispatching mechanism. Despite its flexibility, ROCR is easy to use, with only three commands and reasonable default values for all optional parameters.

Version: 1.0-7
Depends: gplots, methods
Published: 2015-03-26
Author: Tobias Sing, Oliver Sander, Niko Beerenwinkel, Thomas Lengauer
Maintainer: Tobias Sing <tobias.sing at gmail.com>
License: GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
URL: http://rocr.bioinf.mpi-sb.mpg.de/
NeedsCompilation: no
Citation: ROCR citation info
Materials: README NEWS INSTALL
In views: MachineLearning, Multivariate
CRAN checks: ROCR results

Downloads:

Reference manual: ROCR.pdf
Package source: ROCR 1.0-7.tar.gz
Windows binaries: r-devel: ROCR 1.0-7.zip, r-release: ROCR 1.0-7.zip, r-oldrel: ROCR 1.0-7.zip
OS X Snow Leopard binaries: r-release: ROCR 1.0-7.tgz, r-oldrel: ROCR 1.0-7.tgz
OS X Mavericks binaries: r-release: ROCR 1.0-7.tgz
Old ...

- Seleccionando el paquete en CRAN o buscando con un explorador en la web “CRAN package ROCR”

Primeras nociones con R

- **Ayuda**

<code>help()</code>	Ayuda
<code>?</code>	Ayuda
<code>help.start()</code>	Ayuda en HTML
<code>help.search()</code>	Busca términos relacionados

- Comandos básicos de R: **expresiones y asignaciones**
 - Las **expresiones** devuelven un resultado
 - Las **asignaciones crean un objeto** con el resultado. Se utiliza “<-”, “>-” o “=“
 - Los comandos se pueden separar por “;” y se pueden agrupar con llaves { }
 - Un comando se puede escribir en **varias líneas**
- R es **sensible** a mayúsculas y minúsculas
- El carácter **#** (“almohadilla”) se utiliza para incluir **comentarios**

Objetos y workspace

- R crea y manipula **objetos**
 - Variables, vectores, matrices, ...
 - Objetos que contienen los datos (*data.frame*)
 - Objetos creados por el usuario
 - Resultados de las funciones
- Los **objetos** se guardan en un **workspace (.RData)**
 - “File” / “Load Workspace”
 - “File” / “Save Workspace”
- **Funciones**

<code>load()</code>	Carga un workspace
<code>save.image()</code>	Salva un workspace completo
<code>save()</code>	Salva objetos de un sesión
<code>ls()</code>	Muestra los objetos dentro del workspace
<code>rm()</code>	Borra objetos

Ejemplo: Ayuda

```
> ?mean
starting httpd help server ... done
> help(mean)
> help.search("mean")
> ?stats::weighted.mean
> example("mean")

mean> x <- c(0:10, 50)

mean> xm <- mean(x)

mean> c(xm, mean(x, trim = 0.10))
[1] 8.75 5.50
> search()
[1] ".GlobalEnv"           "package:stats"      "package:graphics"   "package:grDevices"
[5] "package:utils"         "package:datasets"    "package:methods"    "Autoloads"
[9] "package:base"
> library(survival)
Mensajes de aviso perdidos
package 'survival' was built under R version 3.1.2
> search()
[1] ".GlobalEnv"           "package:survival"  "package:stats"      "package:graphics"
[5] "package:grDevices"    "package:utils"     "package:datasets"    "package:methods"
[9] "Autoloads"            "package:base"
```

- La forma más común de pedir ayuda sobre una función de R es teclear **?función**

Ejemplo: Expresiones y asignaciones

```
> 6 + 3
[1] 9
> a <- 6 + 3
> a
[1] 9
> A
Error: object 'A' not found
> A <- 3
> a
[1] 9
> A
[1] 3
> b <- 3 ; c = 4
> d <-
+ 6
> b
[1] 3
> c
[1] 4
> d
[1] 6
> ls()
[1] "a"   "A"   "b"   "c"   "d"   "x"   "xm"
> rm(b, c, d)
> ls()
[1] "a"   "A"   "x"   "xm"
```

Entorno y directorios de trabajo

source()	Ejecuta los comandos que están guardados en un fichero
getwd()	Muestra el directorio de trabajo
setwd()	Cambia el directorio de trabajo

- Escribiendo el nombre de directorios y ficheros
 - `setwd("C:/Ejemplo/Otro")`
 - `setwd("C:\\Ejemplo\\Otro")`
- “File” / “Change Dir” : cambia el directorio de trabajo
- “File” / “Save History” : salva todos los comandos de R de una sesión
- Es habitual salvar las funciones creadas por el usuario en un script de R, y cargarlas con **source()** en el script donde las vamos a usar

Ejemplo: Workspace y directorio de trabajo

```
> ## Salvar todo
> save.image("C:/Bioestadistica con R/Temp/Ejemplo.RData")

> ## Salvar un objeto
> save( a, file="C:/Bioestadistica con R/Temp/Ejemplo2.RData")

> load("Ejemplo.RData")
> ls()
[1] "a"   "A"   "x"   "xm"

> getwd()
[1] "C:/windows/system32"
> setwd("C:/Bioestadistica con R/Temp")
> getwd()
[1] "C:/Bioestadistica con R/Temp"
> setwd("C:/Bioestadistica con R")
> getwd()
[1] "C:/Bioestadistica con R"
```

Software relacionado con R

- **Editor de R: Tinn-R (Windows)**
 - Página web: [www.sciviews.org/Tinn-R](http://nbcgib.uesc.br/lec/software/editores/tinn-r/en), pero se ha movido a <http://nbcgib.uesc.br/lec/software/editores/tinn-r/en>
- **RStudio**
 - Página web: www.rstudio.com
 - Es un **entorno de desarrollo** muy potente relacionado con R
- **R Commander** usa un **interfaz gráfico** con ventanas para ejecutar R
 - R Commander no es un software: se instala con la librería “**Rcmdr**”
 - R Commander permite: abrir ficheros de datos, ejecutar análisis estadísticos y realizar gráficos mediante ventanas. Se puede guardar la sintaxis de R

Ejercicios

- Instalar Tinn-R, si se desea: <http://sciviews.org/Tinn-R/>
- Instalar los siguientes **paquetes de R**:
 - foreign, xlsx, nortest, rrcov, KernSmooth, corrplot, pROC, survivalROC, MASS, ICC, rms, nlme, car, ResourceSelection
 - qvalue de Bioconductor
 - source("http://bioconductor.org/biocLite.R")
 - biocLite("qvalue")
- **Workspace**
 - Abrir R, crear algunos objetos sencillos
 - Salvar un workspace
 - Cerrar R
 - Abrir el workspace salvado

Estadística Aplicada a la Investigación Biomédica con R

2 Variables y Objetos en R

- ✓ **Vectores**
- ✓ **Funciones básicas**
- ✓ **Arrays y matrices**
- ✓ **Listas**
- ✓ **Fórmulas**

Vectores

- **Estructura o conjunto de datos indexados**
 - Todos los datos son del mismo tipo: numérico, carácter, lógico (TRUE/FALSE), ...
- **Asignación** de valores a un vector
 - Funciones: concatenar `c()` y `assign()`
- **Indexación de vectores**
 - Para seleccionar uno o varios elementos se usan los corchetes []
- **Aritmética de vectores**
 - El operador se aplica a los elementos del vector
 - Se usa el reciclaje de vectores de distinta longitud: repite los valores del vector de menor longitud

Ejemplo: Vectores numéricos

```
> a <- c(1,5,7,8)
> a
[1] 1 5 7 8
> assign("b", c(9,5,3,2))
> b
[1] 9 5 3 2
> a+b
[1] 10 10 10 10
>
> a[2]
[1] 5
> a[5]
[1] NA
>
> c<-c(4,6,9)
> a+c
[1] 5 11 16 12
Warning message:
In a + c : longer object length is not a multiple of shorter object length
>
```

- NA = “Not Available”, Missing
- En la suma de vectores de distinta longitud, el vector c que es (4,6,9) se recicla a (4,6,9,4)

Secuencias regulares

- **Operador “:”**
 - Hace referencia a todos los números enteros en un rango
- **Funciones**
 - *seq(n1, n2, by=salto)*
 - Hace referencia a todos los números entre n1 y n2, sumando una determinada cantidad
 - *rep(vector, num.veces)*
 - Repite un vector o variable el número de veces indicado

Ejemplo: Secuencias

```
> 1:20
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
> n<-5
> 1:n
[1] 1 2 3 4 5
> 1:(n-1)
[1] 1 2 3 4
>
> seq(1,10, by=2)
[1] 1 3 5 7 9
>
> rep(1:3, 2)
[1] 1 2 3 1 2 3
>
> mean.res <- rep(NA,20)
> mean.res
[1] NA NA
```

- La función **rep()** se puede utilizar para crear vectores con NA (missings) que luego se irán cargando con resultados

Operadores aritméticos y lógicos. Funciones básicas

+	Suma
-	Resta
*	Multiplicación
/	División
^	Potencia
**	Potencia

<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
==	Igualdad
!=	Desigualdad
&	Y (AND)
	O (OR)
!	Negación lógica

<i>sqrt()</i>	Raíz cuadrada
<i>log()</i>	Logaritmo natural
<i>log10()</i>	Logaritmo decimal
<i>exp()</i>	Exponencial
<i>round()</i>	Redondea con un nº de decimales
<i>signif()</i>	Muestra decimales significativos
<i>trunc()</i>	Trunca al entero inferior
<i>ceiling()</i>	Entero superior
<i>sin(), cos(), tan()</i>	Funciones trigonométricas: seno, coseno, tangente
<i>asin, acos, atan</i>	Arcoseno, arcocoseno, arcotangente

- Los **operadores y las funciones básicas** tienen **carácter vectorial**, es decir, si se aplican a un vector generan otro vector
- Se aplican componente a componente

Funciones básicas de vectores

<i>max()</i>	Máximo
<i>min()</i>	Mínimo
<i>range()</i>	Rango
<i>length()</i>	Longitud
<i>sum()</i>	Suma
<i>prod()</i>	Producto
<i>mean()</i>	Media
<i>median()</i>	Mediana
<i>sd()</i>	Desviación típica
<i>var()</i>	Varianza

<i>sort()</i>	Ordena
<i>order()</i>	Indica la posición de los elementos ordenados
<i>rank()</i>	Asigna rangos

<i>which.max()</i>	Posición del máximo
<i>which.min()</i>	Posición del mínimo
<i>which()</i>	Posición de los elementos de un vector que cumplen una condición

- El operador ***%in%*** se usa para comprobar si determinados valores existen dentro de un vector

Ejemplo: Funciones básicas

```
> x<-c(3.3, 3.2, 1.7, 2.3, 4.5, 2.6)
> min(x)
[1] 1.7
> max(x)
[1] 4.5
> which.min(x)
[1] 3
> which.max(x)
[1] 5
> which( x == 3.2 )
[1] 2
> which( x > 3 )
[1] 1 2 5
> c(1.7, 1.8) %in% x
[1] TRUE FALSE
> sum(x)
[1] 17.6
> mean(x)
[1] 2.933333
> sd(x)
[1] 0.9688481
> sort(x)
[1] 1.7 2.3 2.6 3.2 3.3 4.5
> order(x)
[1] 3 4 6 2 1 5
> rank(x)
[1] 5 4 1 2 6 3
```

Missings

- Se llaman **Missings** a los **valores desaparecidos o desconocidos**
- En R se utiliza **NA** (Not Available) para manejar los missings
- Algunas funciones que se aplican a un vector con algún NA, dan como **resultado NA**
- La función ***is.na()*** se usa para saber si un número o qué elementos de un vector son missing
- También hay variables que tienen valor **NaN** (Not a Number). Resultado de operaciones

Ejemplo: Missings

```
> ## Missing
> x <- c( 3, NA, 6, 2, 4, 6, NA, 1)
> x
[1] 3 NA 6 2 4 6 NA 1
> is.na(x)
[1] FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE
> sum(is.na(x))      ## Número de missings en x
[1] 2
> mean(x)
[1] NA
> mean(x, na.rm=T)
[1] 3.666667
```

- Los **vectores lógicos** contienen TRUE / FALSE en cada componente
- Internamente, se pueden tratar como un vector numérico:
 - **FALSE = 0**
 - **TRUE = 1**

Caracteres

- Los vectores de caracteres se usan para almacenar etiquetas
- Se usa “ ” o ‘ ’ (comillas dobles o comillas simples)
- Carácter de escape “\”
 - “\n” salto de línea
 - “\t” tabulador
- Funciones de manipulación de caracteres

<i>paste()</i>	Concatena varias cadenas
<i>cat()</i>	Combina caracteres y los muestra por pantalla o en ficheros
<i>print()</i>	Muestra por pantalla
<i>substring()</i>	Extrae una parte del texto
<i>grep()</i>	Busca un subtexto en un texto

Ejemplo: Caracteres

```
> a <- c("Hola", "Adiós")
> b <- "Hoy es un buen día"
> length(a)
[1] 2
>
> paste( "Me dijo", a[1], "y yo le respondí", a[2])
[1] "Me dijo Hola y yo le respondí Adiós"
> paste( "Me dijo", a[1], "y yo le respondí", a[2], sep="-" )
[1] "Me dijo-Hola-y yo le respondí-Adiós"
> paste( "Me dijo", a[1], "y yo le respondí", a[2], sep="" )
[1] "Me dijoHolay yo le respondíAdiós"
>
> grep("es",b)
[1] 1
> grep("xx",b)
integer(0)
> substring ( "Me dijo Hola" , 4, 7 )
[1] "dijo"
>
> print( a, b )
Error in print.default(a, b) : invalid 'digits' argument
In addition: Warning message:
In print.default(a, b) : NAs introduced by coercion
> print(a)
[1] "Hola"  "Adiós"
> cat( a, b )
Hola Adiós Hoy es un buen día> cat( a, b, "\n")
Hola Adiós Hoy es un buen día
```

Indexación de vectores

- Permite seleccionar un **subconjunto** de los elementos del vector:
 - Se usan corchetes []
 - Genera otro vector con aquellos elementos seleccionados
- Vectores enteros positivos
 - Los elementos indicados son seleccionados
- Vectores enteros negativos
 - Los elementos indicados no son seleccionados, son excluidos
- Vectores lógicos
 - Misma longitud que el vector
 - Se aplica a cada elemento: si es TRUE se selecciona, si FALSE no se selecciona

Ejemplo: Indexación de vectores

```
> a <- c (3,2,1,2,5,6,NA,4,5,NA,4,8,NA,3,7)
> a[1]
[1] 3
> a[c(1,3,8)]
[1] 3 1 4
> a[4:7]
[1] 2 5 6 NA
> a[a>4]
[1] 5 6 NA 5 NA 8 NA 7
> is.na(a)
[1] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE
FALSE FALSE
> a[is.na(a)==FALSE]
[1] 3 2 1 2 5 6 4 5 4 8 3 7
> a[!is.na(a)]
[1] 3 2 1 2 5 6 4 5 4 8 3 7
> a[!is.na(a) & a>4]
[1] 5 6 5 8 7
> a[-(1:2)]
[1] 1 2 5 6 NA 4 5 NA 4 8 NA 3 7
>
```

Arrays y matrices

- Un **array** es un conjunto de datos de **k dimensiones**
 - Vector, cuando $k=1$
 - **Matriz**, cuando $k=2$
 - Todos los datos son del mismo tipo (numéricos, enteros, carácter, ...)
- **Indexación de arrays**
 - Genera un array con los elementos seleccionados
 - Se usa **[] con comas** para cada dimensión. Por ejemplo: **[,] o [, ,]**
 - Si se deja un **espacio en blanco**, se seleccionan todos los elementos de esa dimensión
 - El nuevo array puede tener **menos dimensiones**
 - **Vector de índices** para cada dimensión
 - **Vectores enteros** positivos y negativos (elementos seleccionados o no)
 - **Vectores lógicos**: elementos seleccionados, los que cumplen la condición (TRUE)

Ejemplo: Indexación de arrays

```
> a <- 1:24
> a
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
> dim(a) <- c(4,6)
> a
 [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1     5     9    13    17    21
[2,]    2     6    10    14    18    22
[3,]    3     7    11    15    19    23
[4,]    4     8    12    16    20    24
> a[1,2]
[1] 5
> a[1,]
[1] 1 5 9 13 17 21
```

Arrays y matrices

- La función **array()** se usa para crear arrays
 - $x <- \text{array}(\text{valores}, \text{dim}=c(,,))$
 - Se usa el reciclaje de valores
- La función **matrix()** se usa para crear matrices
 - $x <- \text{matrix}(\text{valores}, \text{num.filas}, \text{num.col})$
- Funciones para matrices

dim()	Dimensión de un array o matriz
nrow()	Número de filas de una matriz
ncol()	Número de columnas de una matriz
t()	Traspuesta de una matriz
diag()	Diagonal de una matriz
rbind()	Une vectores por filas para construir una matriz
cbind()	Une vectores por columnas para construir una matriz

Ejemplo: Arrays y matrices

```
> x <- array ( NA, c(2,5))
> x
[,1] [,2] [,3] [,4] [,5]
[1,]    NA    NA    NA    NA    NA
[2,]    NA    NA    NA    NA    NA
> y <- array ( 1:2, c(2,2,2))
> y
, , 1

[,1] [,2]
[1,]    1    1
[2,]    2    2

, , 2

[,1] [,2]
[1,]    1    1
[2,]    2    2
```

Ejemplo: Arrays y matrices

```
> x <- matrix ( 1:10, 2, 5 )
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> dim(x)
[1] 2 5
> nrow(x)
[1] 2
> ncol(x)
[1] 5
> t(x)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
> diag(x)
[1] 1 4
> rbind(1:4, 6:9, 10:13)
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    6    7    8    9
[3,]   10   11   12   13
```

Listas

- Una **lista** es un objeto que consiste en una **colección ordenada de objetos**, que se llaman **componentes**
 - Los componentes de la lista pueden ser objetos de **diferentes clases**: arrays, vectores, variables, de diferentes tipo y dimensión
 - Se crean con ***list()***
 - Cada componente se accede con dobles corchetes **[[]]** . Si se han nombrado los componentes, se pueden acceder con **\$**
- Una de las utilidades de las listas es agrupar los **resultados de una función**
- Funciones para listas

<i>length()</i>	Devuelve el número de componentes de la lista
<i>names()</i>	Asigna nombres a los componentes de la lista o los muestra
<i>c()</i>	Concatena listas

Ejemplo: Listas

```
> a <- 4
> b <- c(1,2,3)
>
> list1 <- list ( a, b, "Hola")
> list1
[[1]]
[1] 4

[[2]]
[1] 1 2 3

[[3]]
[1] "Hola"

> list1[[1]]
[1] 4
> list1[[2]]
[1] 1 2 3
> list1[[2]][3]      ## 3º elemento del 2º componente
[1] 3
```

Ejemplo: Listas

```
> names(list1) <- c("var", "vector", "texto")
> list1
$var
[1] 4

$vector
[1] 1 2 3

$texto
[1] "Hola"

> list1$vector
[1] 1 2 3
>
> ## Alternativa (así se usa con los resultados de las funciones)
> list2 <- list(var=a, vector=b, texto="Hola")
> list2
$var
[1] 4

$vector
[1] 1 2 3

$texto
[1] "Hola"
```

Funciones de conversión y verificación de tipos

<i>is.numeric()</i>	Es numérico
<i>is.integer()</i>	Es entero
<i>is.character()</i>	Es carácter
<i>is.matrix()</i>	Es matriz
<i>is.data.frame()</i>	Es data frame
<i>is.vector()</i>	Es vector
<i>is.factor()</i>	Es factor
<i>is.na()</i>	Es NA (missing)
<i>is.infinite()</i>	Es infinito

<i>as.numeric()</i>	Convierte a numérico
<i>as.integer()</i>	Convierte a entero
<i>as.character()</i>	Convierte a carácter
<i>as.matrix()</i>	Convierte a matriz
<i>as.data.frame()</i>	Convierte a data frame
<i>as.vector()</i>	Convierte a vector
<i>as.factor()</i>	Convierte a factor
<i>as.formula()</i>	Convierte a fórmula

- No confundir con otras funciones como *numeric(x)*, *integer(x)*, *character(x)*, que crean un vector del tipo elegido con longitud x

Modelos en R: fórmulas

- Una fórmula es una **expresión simbólica** que define la forma estructural de **un modelo**
- $Y \sim X$ significa que **Y es modelada por X** donde **Y** es la **variable respuesta** y **X** describe cómo **las variables predictoras** son usadas
 - $Y \sim X_1 + X_2$ varias variables independientes
 - $Y \sim X_1 * X_2$ para modelar una **interacción**, incluye los términos X_1 y X_2
 - $Y \sim X_1:X_2$ incluye solo el término de la interacción (no se suele usar)
 - $Y \sim I(X_1 * X_2)$ La **función *I()*** se usa para operaciones entre variables
 - $Y \sim 1$ Modelo con solo el **intercept** (modelo nulo)
 - $Y \sim X - 1$ Modela X sin el intercept
- La función ***as.formula()*** se utiliza para convertir un texto en una formula
 - Permite la **construcción dinámica** de los modelos que se van a ajustar

Ejercicios

- Crear un **vector** “a” de 10 valores cualesquiera con números entre 1 y 100
 - Crear 2 subvectores de “a” con los números que están en las posiciones impares y pares (con `seq()` se pueden localizar los pares e impares)
 - Crear un subvector con los números de “a” mayores que 30
- Supongamos que tenemos 3 variables calculadas con los valores de un OR y su intervalo de confianza: $OR=1.33$, $OR_L=1.09$ y $OR_U=1.96$.
 - Utiliza la función **`paste()`** para generar el siguiente **texto**:
`OR=1.33, IC95%:1.09 - 1.96`
- Define una **matriz** de dimensiones 30x3 cargada con los números 1 al 90
 - Crea un subvector con la 2º columna y las 10 primeras filas
 - Calcula la longitud y la media de ese subvector

Estadística Aplicada a la Investigación Biomédica con R

3 Manejo de Datos. Ficheros y *data frames*

- ✓ ***Data frames***
- ✓ **Factores**
- ✓ **Lectura e importación de ficheros**
- ✓ **Ficheros de salida**

Data frame

- Un ***data frame*** es el **objeto** donde se **almacenan los datos** para analizar
 - Es un **conjunto de variables** con la misma longitud
 - Se crean con ***data.frame()***
 - Algunas funciones de **lectura de ficheros** crean directamente un data frame
 - Las **filas** son las **observaciones**, los individuos
 - Las **columnas** son las **variables**. Todas las columnas tienen **un nombre**
 - Cada columna o variable es un **vector** que se denomina ***df\$columna***
 - Se pueden añadir nuevas variables, con ***df\$columna <- vector de valores***
 - Se puede acceder también a una parte del dataframe con los índices entre corchetes **[,]**, donde la 1º dimensión son las observaciones, y la 2º las variables
 - Se puede **acceder a las variables** por el nombre o por la posición de la columna

Ejemplo: *data frame*

```
> df <- data.frame( ID=c(1,3,4,5,8,9,10,11), edad=c(34,46,23,19,23,11,14,34),  
+                     sexo=c("H","M","M","M","H","M","H","M") )  
> df  
ID edad sexo  
1 1 34 H  
2 3 46 M  
3 4 23 M  
4 5 19 M  
5 8 23 H  
6 9 11 M  
7 10 14 H  
8 11 34 M  
> df$edad  
[1] 34 46 23 19 23 11 14 34  
> mean(df$edad)  
[1] 25.5  
> df[, c("edad", "sexo")]  
edad sexo  
1 34 H  
2 46 M  
3 23 M  
4 19 M  
5 23 H  
6 11 M  
7 14 H  
8 34 M
```

Ejemplo: *data frame*

```
> df[, 2:3]
   edad sexo
1   34   H
2   46   M
3   23   M
4   19   M
5   23   H
6   11   M
7   14   H
8   34   M
> names(df)
[1] "ID"    "edad"   "sexo"
>
> df$tratamiento <- c(0,0,0,1,1,0,1,0)
> df
   ID edad sexo tratamiento
1  1   34   H         0
2  3   46   M         0
3  4   23   M         0
4  5   19   M         1
5  8   23   H         1
6  9   11   M         0
7 10   14   H         1
8 11   34   M         0
>
```

Data frame

- Funciones para *data frames*

<code>head()</code>	Muestra las primeras filas del <i>data frame</i> o matriz
<code>tail()</code>	Muestra las últimas filas del <i>data frame</i> o matriz
<code>names()</code>	Vector que contiene los nombres de las variables, columnas
<code>na.omit()</code>	Crea una dataframe eliminando todas las filas que contienen algún NA

- Son parecidas a una **matriz**, y algunas de las **funciones para matrices** se pueden utilizar con los data frames (***dim*, *nrow*, *ncol*, *cbind*, *rbind*,**)

Ejemplo: *data frame*

```
> dim(df)
[1] 8 3
> head(df)
  ID edad sexo tratamiento
1  1   34   H         0
2  3   46   M         0
3  4   23   M         0
4  5   19   M         1
5  8   23   H         1
6  9   11   M         0
> names(df)
[1] "ID"          "edad"        "sexo"        "tratamiento"
> names(df)[2]
[1] "edad"
>
```

Data frame: función merge()

- La función **merge()** se utiliza para **unir data frames** uniendo **sus variables** (columnas)
 - El **resultado** de la función *merge()* es **otro data frame** (*xx <- merge (...)*)
- *merge(x, y, by = intersect(names(x), names(y)), by.x = by, by.y = by, all = FALSE, all.x = all, all.y = all, sort = TRUE, ...)*
 - Une los *data frame* x e y en función de **una o varias columnas** que se especifican en **by, by.x o by.y**. Se utiliza **by** si los nombres coinciden
 - Los nombres de las columnas se ponen con comillas “”
 - Las opciones **all, all.x, all.y** se utilizan para indicar qué filas de los *data frames* van a formar parte del *data frame* “resultado”
 - La función *merge()* solo une 2 dataframes
- Para **unir observaciones** (filas) de 2 *data frames* que contienen las mismas columnas (variables), se puede utilizar **rbind()**

Ejemplo: función *merge()*

```
> df1 <- data.frame( ID =c(1,2,3,4), edad=c(12,34,44,54) )
> df2 <- data.frame( ID2=c(1,2,3,4), hta=c(0,1,0,1) )
> df3 <- merge(df1, df2, by.x="ID", by.y="ID2" )
> df3
   ID  edad  hta
1  1    12    0
2  2    34    1
3  3    44    0
4  4    54    1
>
> df2 <- data.frame( ID2=c(1,2,4,5), hta=c(0,1,0,1) )
> df3 <- merge(df1, df2, by.x="ID", by.y="ID2" )
> df3
   ID  edad  hta
1  1    12    0
2  2    34    1
3  4    54    0
> df3 <- merge(df1, df2, by.x="ID", by.y="ID2", all=T )
> df3
   ID  edad  hta
1  1    12    0
2  2    34    1
3  3    44    NA
4  4    54    0
5  5    NA    1
```

Ejemplo: función *merge()*

```
> df3 <- merge(df1, df2, by.x="ID", by.y="ID2", all.x=T )
> df3
  ID edad hta
1  1    12   0
2  2    34   1
3  3    44   NA
4  4    54   0
> df3 <- merge(df1, df2, by.x="ID", by.y="ID2", all.y=T )
> df3
  ID edad hta
1  1    12   0
2  2    34   1
3  4    54   0
4  5    NA   1
>
> ## Uniendo filas con rbind
> df1 <- data.frame( ID =c(1,2,3,4), edad=c(12,34,44,54) )
> df2 <- data.frame( ID =c(5,6), edad=c(42,28) )
> df3 <- rbind(df1, df2)
> df3
  ID edad
1  1    12
2  2    34
3  3    44
4  4    54
5  5    42
6  6    28
```

Factor

- Se utiliza para indicar que un vector es una **variable categórica**
 - Los factores tienen diferentes **categorías o niveles (*levels*)**
 - La función ***levels()*** muestra los niveles de un factor, y también se puede usar para **asignar** los niveles
 - Los vectores de **tipo carácter**, por defecto son factores
 - La variables categóricas tienen un **tratamiento especial** en muchas técnicas estadísticas
 - Por ejemplo, en los modelos de regresión se tratan con **variables dummy** y R crea esas variables dummy automáticamente para los factores
- Cuando se lee un fichero de datos por primera vez, se debe indicar qué variables son **categóricas**, definiéndolas como factores
 - Utilizar ***as.factor()*** o ***factor()***

Ejemplo: Factores

```
> df <- data.frame( ID=c(1,3,4,5,8,9,10,11), edad=c(34,46,23,19,23,11,14,34),  
+                     sexo=c("H","M","M","M","H","M","H","M") )  
> df$tratamiento <- c(0,0,0,1,1,0,1,0)  
> is.factor(df$edad)  
[1] FALSE  
> is.factor(df$sexo)  
[1] TRUE  
> is.factor(df$tratamiento)  
[1] FALSE  
>  
> df$sexo  
[1] H M M M H M H M  
Levels: H M  
> df$tratamiento  
[1] 0 0 0 1 1 0 1 0  
> levels(df$tratamiento)  
NULL  
>  
> df$tratamiento <- as.factor(df$tratamiento)  
> df$tratamiento  
[1] 0 0 0 1 1 0 1 0  
Levels: 0 1  
> levels(df$tratamiento)  
[1] "0" "1"  
>
```

Ejemplo: Problemas con factores

```
> ## Algunos problemas con Factores
> x <- c(0,0,0,0,0,1,1,1,1,1,1,1,1,2,2,2,2)
> x<-factor(x)
> table(x)
x
0 1 2
5 9 4
> levels(x)
[1] "0" "1" "2"
> ## Asigna valores 1,2,3 .... a las categorías ordenadas
> as.integer(x)
[1] 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3
> x.num <- as.integer( as.character(x) )      ## Truco
> x.num
[1] 0 0 0 0 0 1 1 1 1 1 1 1 1 2 2 2 2
> ## Guarda los levels siempre
> x [ x == 2 ] <- 1
> x
[1] 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
Levels: 0 1 2
> table(x)
x
0 1 2
5 13 0
> x = as.factor ( as.character( x ) )
> x
[1] 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
Levels: 0 1
```

Lectura de ficheros tabulados: *read.table*

- La función ***read.table()*** se utiliza para **leer ficheros de texto**
 - Devuelve un ***data frame***
- *read.table(file, header = FALSE, sep = "", quote = "\"\"", dec = ".", row.names, col.names, as.is = !stringsAsFactors, na.strings = "NA",)*
- **Parámetros** más usados:
 - file ruta completa del fichero
 - header si la 1º línea incluye los nombres de las variables (columnas)
 - sep carácter que separa las columnas ("\t", ",", ";", ...)
 - quote indica si los caracteres van entre comillas
 - dec carácter para separación decimal

Lectura de ficheros

- La función ***read.csv()*** se utiliza para leer ficheros de texto separados por comas
 - Se usa como ***read.table()***
 - Conviene especificar el carácter de separación (coma o punto y coma)
 - Otra función parecida es ***read.delim()***
- La función ***data()*** permite leer conjuntos de datos que ya existen en formato R
 - Se suele utilizar con los ficheros de datos que acompañan como **ejemplos** a los **paquetes** de R

Ejemplo: Lectura de ficheros

```
> f1 <- read.table(file="C://Bioestadistica con R/Ficheros para importar/Ejemplo 1.txt",
+                     header=T)
> dim(f1)
[1] 12  3
> head(f1)
  id sexo edad
1  1     1   24
2  2     1   39
3  3    NA   33
4  4     1   NA
5  5     2   25
6  6     2   26
>
> f2 <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
> dim(f2)
[1] 189  10
> head(f2)
  id bajo_pes edad      peso raza fumador part_pre hta irr_urin visi_med
1  4          1   28 54.43164    3       1       1   0       1       0
2 10         1   29 58.96761    1       0       0   0       1       2
3 11         1   34 84.82264    2       1       0   1       0       0
4 13         1   25 47.62769    3       0       1   1       0       0
5 15         1   25 38.55575    3       0       0   0       1       0
6 16         1   27 68.03955    3       0       0   0       0       0
>
```

Importación de ficheros de otros software

- Para importar ficheros de otros **paquetes estadísticos** se usa la librería ***foreign***
- Funciones de lectura de los paquetes estadísticos más importantes
 - **SAS** ***read.xport()***
 - **STATA** ***read.dta()*** ***write.dta()***
 - **SPSS** ***read.spss()***
 - Otros paquetes: EPIINFO, MINITAB, S-PLUS, SYSTAT
- Hay varias funciones y librerías para abrir **hojas Excel**
 - La librería ***xlsx*** funciona muy bien, con las nuevas versiones de Excel y también con las antiguas. La función de lectura es ***read.xlsx()***
 - Hay que instalar la librería de R ***rJava*** e instalar la versión de **Java 64-bits**
 - Para leer versiones antiguas de Excel, se usaba la función ***read.xls()*** de la librerías ***gdata*** o ***xlsReadWrite***

Ejemplo: Importación de ficheros

```
> library(foreign)
> ## SPSS
> f3<-read.spss("C://Bioestadistica con R/Ficheros para importar/Ejemplo 3.sav",
+                  to.data.frame=TRUE)
> head(f3)
  id sexo edad
1 1     2   34
2 2    NA   41
3 3     1   NA
4 4     2   48
5 5     2   32
6 6     1   21
>
> ## STATA
>
> f4<-read.dta("C://Bioestadistica con R/Ficheros para importar/Ejemplo 4.dta")
> head(f4)
  id sexo edad
1 1     1   54
2 2     2   44
3 3    NA   43
4 4     1   33
5 5     1   21
6 6     2   26
```

Ejemplo: Importación de ficheros de Excel

```
> ## EXCEL - 2 hojas
> library(xlsx)
>
> f5<-read.xlsx ( "C://Bioestadistica con R/Ficheros para importar/Ejemplo 5.xlsx",
+                     sheetIndex=1 )
> head(f5)
  id edad
1 1   33
2 2   15
3 3   25
4 4   17
5 5   26
6 6   17
> f6<-read.xlsx ( "C://Bioestadistica con R/Ficheros para importar/Ejemplo 5.xlsx",
+                     sheetIndex=2 )
> head(f6)
  cod sexo peso altura
1 1   1   47   167
2 2   1   59   145
3 3   0   79   187
4 4   0   93   177
5 5   0   47   156
6 6   0   58   152
```

- El parámetro *sheetIndex=n* controla la hoja del fichero Excel que se desea importar

Ficheros de salida: *write.table*

- La función ***write.table()*** se utiliza para **escribir ficheros de texto**
 - Escribe un **objeto** de R de una vez. Normalmente una matriz o un data frame
- *write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ", eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = TRUE,)*
- Opciones más usadas:
 - **x** objeto de R que se quiere escribir
 - **file** ruta completa del fichero
 - **append** se añade a un fichero ya existente
 - **sep** carácter de separación
 - **quote** indica si los caracteres van entre comillas
 - **eol** carácter que separa las filas ("\n" es salto de línea)
 - **row.names** incluye nombre de las filas
 - **col.names** incluye nombre de las columnas

Ficheros de salida

- Una alternativa para escribir resultados:
 - Abrir una conexión con un fichero con la función ***file()***
 - Escribir cada línea con la función ***cat()***
 - Cerrar el fichero con la función ***close()***
- ***conn <- file(description = "", open = "", blocking = TRUE,)***
 - conn nombre de la conexión que abre
 - description nombre del fichero o de una url
 - open modo de conexión: “w” = write, “r” = read, “a” = appending
- ***cat(... , file = "", sep = " ", fill = FALSE, labels = NULL, append = FALSE)***
 - ... objeto o texto que se desea escribir
 - file conexión
 - sep carácter de separación con otro *cat*

Ejemplo: Ficheros de salida

```
> ## Salvamos a fichero de texto un dataframe
> write.table ( f4, "C://Bioestadistica con R(Temp/Ejemplo 4.txt",
+                 quote=FALSE , sep="\t", col.names=TRUE, row.names=FALSE)
>
> ## Se crea un fichero de salida
> FileOut <- file("C://Bioestadistica con R(Temp/Resultados 4.csv", "w")
>
> cat ( "Variable;Media;SD;Median", file=FileOut, sep="\n")
> cat ( paste ("Edad", mean(f4$edad, na.rm=T), sd(f4$edad, na.rm=T),
+                 median(f4$edad, na.rm=T), sep=";"))
+     , file=FileOut, sep="\n")
>
> close(FileOut)
```

Ejercicios

- Leer los ficheros de texto “Ejemplo 7.csv” y “Ejemplo 8.csv” que están en la carpeta “Ficheros para importar”
 - Definir la variable “fumador” como factor
- Crear en el primer ***data frame*** una variable categorizando la edad (0 para menores de 60 años, 1 para el resto)
- Unirlos con un ***merge()***, creando un *data frame* nuevo que tenga:
 - Todas las variables y observaciones del fichero “Ejemplo 7.txt”
 - El nuevo *data frame* solo incluye la variable “med_2” del fichero “Ejemplo 8.txt”
- Salvar ese nuevo *data frame* en un **fichero de texto** separado por tabuladores (“\t”)

Estadística Aplicada a la Investigación Biomédica con R

4 Gráficos en R

- ✓ **La función *plot***
- ✓ **Gráficos de alto nivel. Parámetros**
- ✓ **Funciones gráficas de bajo nivel**
- ✓ **Salvar gráficos de R**

Gráficos en R

- **Tipos de gráficos en R**
 - **Alto nivel:** crea un nuevo gráfico
 - Gráfico “completo”, es decir, con ejes, etiquetas, ...
 - Siempre inician una nueva ventana para el gráfico, y suelen borrar el anterior
 - **Bajo nivel:** añade información al gráfico
 - Puntos, leyendas, textos, líneas, etiquetas, segmentos, ...
- **Abrir una nueva ventana gráfica de R**
 - *dev.new(), X11(), win.graph()*
- **Salvar los gráficos.** Distintos formatos y calidades, según parámetros
 - *Metafile(), pdf(), png(), jpeg(), bmp(), tiff()*

Función *plot()*

- La función ***plot()*** es una **función genérica**
 - El tipo de gráfico es distinto según el objeto que se le pase
 - Muchos paquetes la utilizan, y le pasan los objetos creados en el paquete
 - Ayuda **?plot** y **?plot.default**
- Ejemplos:
 - *plot(x,y)* es un **gráfico de dispersión** si *x*, *y* son vectores numéricos de la misma longitud
 - *plot(f,x)* es un **boxplot** si *f* es un factor, *y* es un vector
 - *plot(roc.obj)* muestra una curva ROC, siendo *roc.obj* un objeto generado por la función *roc()* del paquete *pROC*, que es la que calcula la curva ROC
 - En este caso, *plot()* es en realidad una llamada a una función que se llama *plot.roc()*

Parámetros

- Las funciones gráficas tienen parámetros
- Control de **parámetros gráficos**
 - Se suelen controlar en las propias **llamadas de las funciones** que realizan el gráfico
 - Los parámetros, dentro de la llamada, pueden estar en el orden que se desee
- La función ***par()*** se usa para cambiar parámetros de forma permanente
 - Afecta a todas las funciones del **gráfico activo**, hasta que se abre un nuevo gráfico, con ***dev.new()***

Parámetros de los gráficos de alto nivel

Parámetro	Descripción	Valores
<i>type</i>	Tipo de gráfico	“p” = puntos “l” = líneas “b” = puntos y líneas “h” = histograma “s” = escalones (supervivencia) “n” = no dibuja nada
<i>xlim, ylim</i>	Límites de los ejes	Rango , xlim=c(0,100)
<i>main</i>	Título del gráfico	Texto
<i>sub</i>	Subtítulo del gráfico	Texto
<i>xlab, ylab</i>	Etiquetas de los ejes	Texto
<i>log</i>	Ejes en escala logarítmica	“x”, “y”, “xy”
<i>axes</i>	Dibuja los ejes	TRUE / FALSE

Otros parámetros gráficos

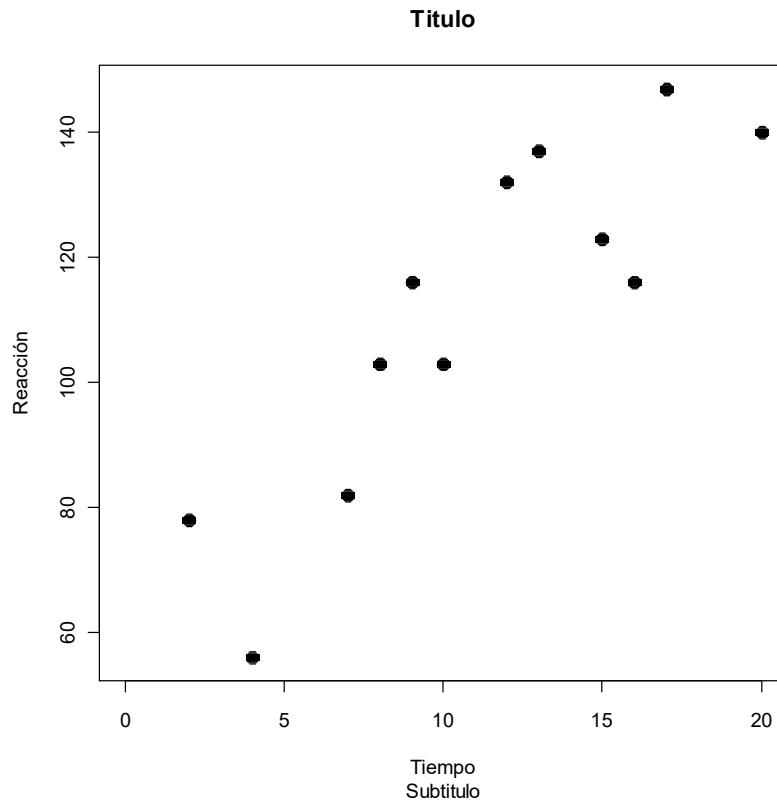
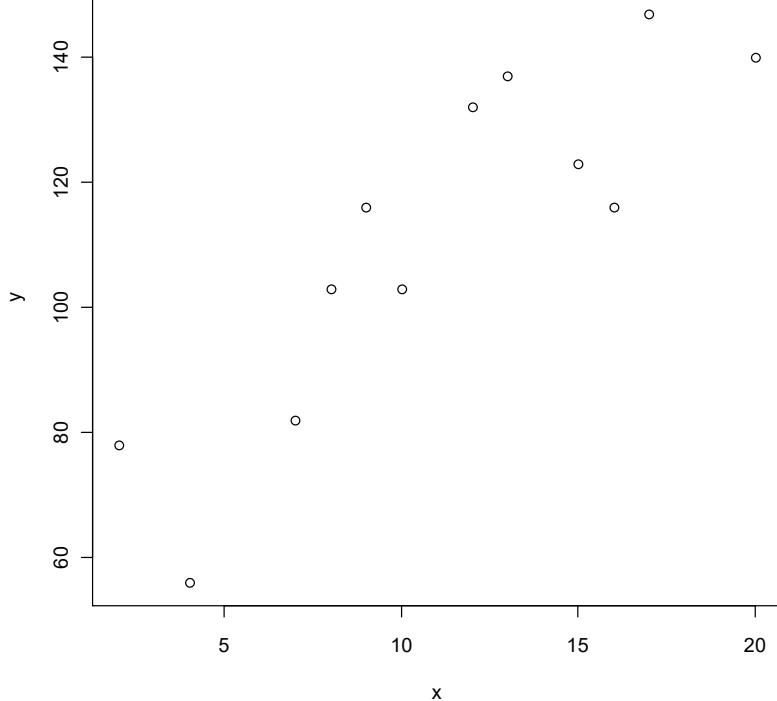
Parámetro	Descripción	Valores
adj	Forma en que los textos son justificados	Rango 0-1 0 = izquierda, 0.5 = centrado, 1 = derecha
ann	Muestra anotaciones : título, subtítulo, etiquetas,...	TRUE / FALSE
font	Fuente a usar para el texto font.axis, font.lab, font.main, font.sub	1=plano, 2=negrita, 3=itálica, 4=negrita-itálica
cex	Escalar tamaño de textos y símbolos cex.axis, cex.lab, cex.main, cex.sub Escalar tamaño de los ejes, las etiquetas, el título y el subtítulo	Valor numérico positivo - El valor por defecto es 1 - Si es > 1 el tamaño será mayor que el normal (cex=1.6) - Si es < 1 el tamaño será menor que el normal (cex=0.8)
col	Color de los símbolos col.axis, col.lab, col.main, col.sub Color de los ejes, las etiquetas, el título y el subtítulo	- Texto, col="red" - Componente RGB, col="#RRGGBB" - Índice, col=25

Otros parámetros gráficos

Parámetro	Descripción	Valores
<i>lty</i>	Tipo de línea	Rango 0-6 0=invisible, 1=sólida, 2=rayas 3=puntos, 4=puntos-rayas, ...
<i>lwd</i>	Ancho de línea	1 por defecto Como el parámetro <i>cex</i>
<i>mfrow</i> <i>mfcol</i>	Prepara la ventana gráfica para varios gráficos independientes. <i>mfrow</i> los rellena por filas, y <i>mfcol</i> por columnas	<i>c(n.row, n.col)</i> indica el número de filas y columnas Ejemplo: <i>par(mfrow=c(2,3))</i>
<i>new</i>	Nuevo gráfico o sobrescribe en el gráfico activo	TRUE / FALSE
<i>add</i>	Añade un gráfico al gráfico activo	TRUE / FALSE Funciona solo en algunos casos
<i>pch</i>	Tipo de símbolo	1-16 o un símbolo entre comillas 15 = cuadrado sólido 16 = círculo sólido
<i>xaxp</i> <i>yaxp</i>	Marcas para los ejes (tick-marks)	<i>c(x1, x2, n)</i> , límites y nº de intervalos

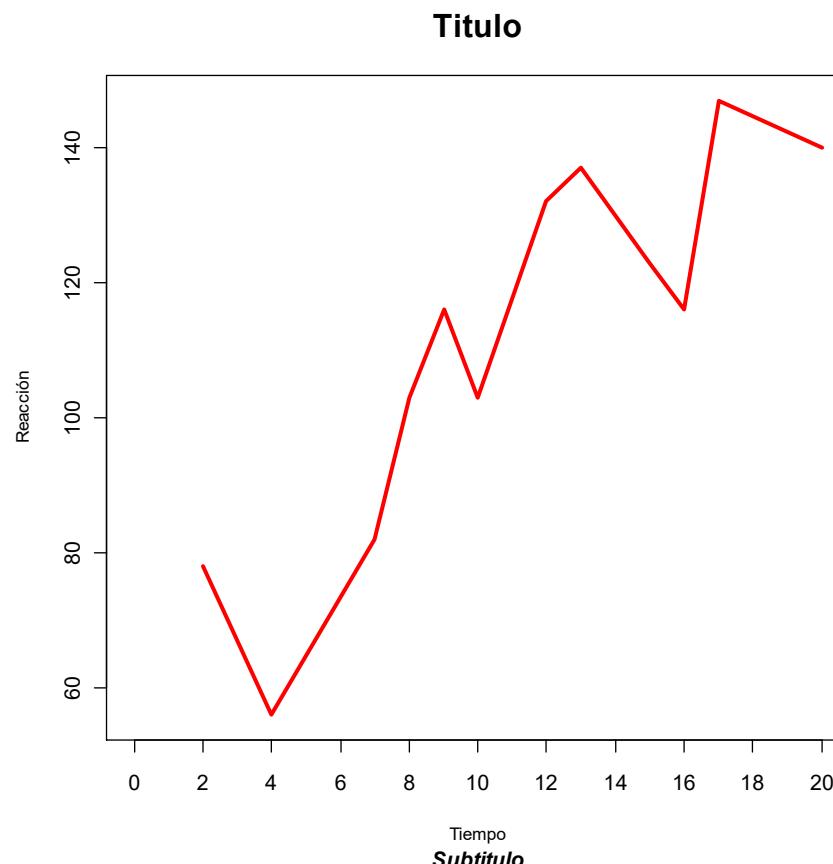
Ejemplo: Gráfico con la función *plot()*

```
> x <- c(2,4,7,8,9,10,12,13,15,16,17,20)
> y <- c(78,56,82,103,116,103,132,137,123,116,147,140)
> ## Gráfico simple
> dev.new()
> plot(x, y)
> ## Gráfico con parámetros
> dev.new()
> plot(x, y, main="Tituto", sub="Subtitulo", xlab="Tiempo", ylab="Reacción",
+       xlim=c(0,20), pch=16, cex=1.5)
```



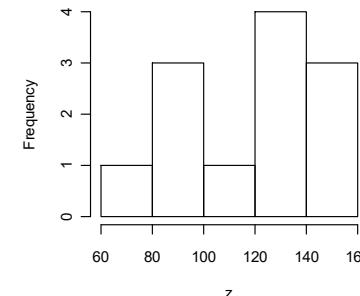
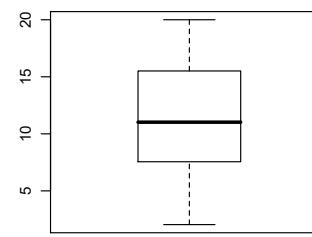
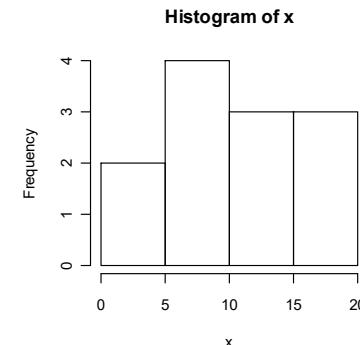
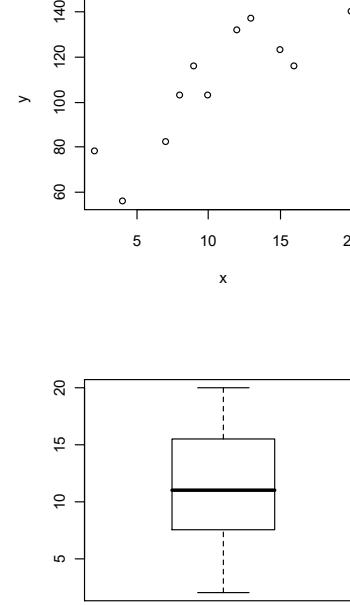
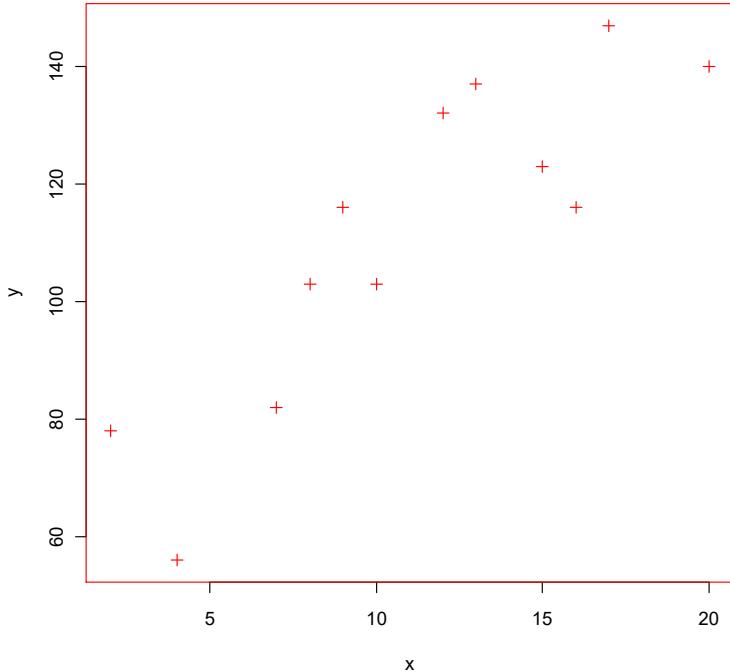
Ejemplo: Gráfico con la función *plot()*

```
> ## Gráfico con parámetros
> dev.new()
> plot(x, y, main="Titulo", sub="Subtitulo", xlab="Tiempo", ylab="Reacción",
+       cex.main = 1.5, cex.lab=0.8, font.sub=4,
+       type="l", lwd=3, col="red",
+       xlim=c(0,20), xaxp=c(0,20,10) )
```



Ejemplo: funciones *par()* y *mfrow()*

```
> dev.new()  
> par(col="red", pch=3)  
> plot(x,y)  
>  
> dev.new()  
> par(mfrow=c(2,2))  
> plot(x,y)  
> hist(x)  
> boxplot(x)  
> hist(y)
```

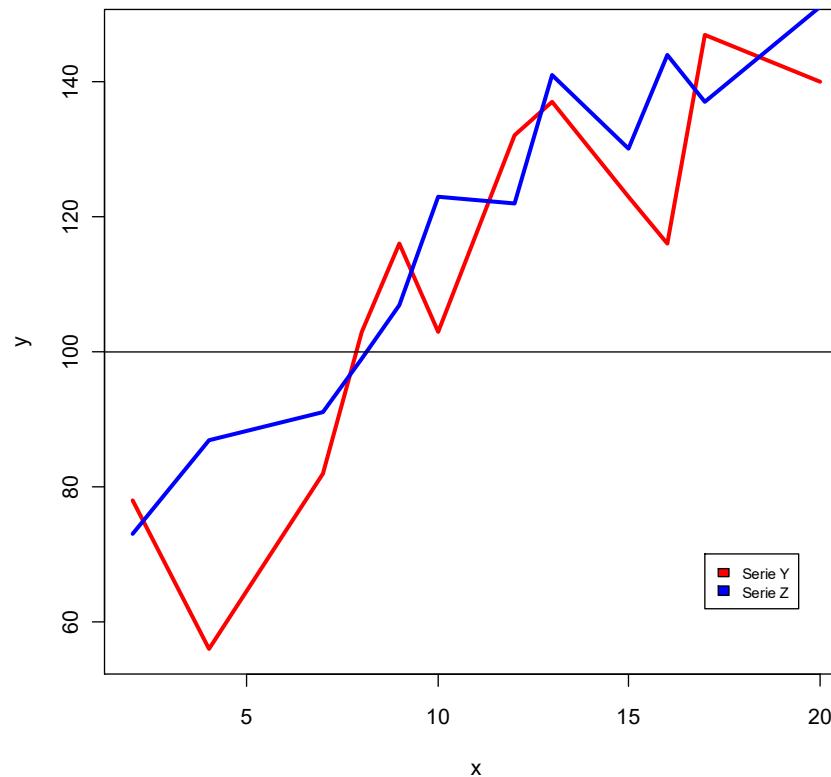


Funciones gráficas de bajo nivel

Parámetro	Descripción	Comentarios
points()	Añade puntos	points(x,y) con x,y vectores <i>pch</i> = tipo de carácter <i>cex</i> = tamaño
lines()	Traza líneas	lines(x,y) con x,y vectores <i>type</i> = tipo de línea <i>lwd</i> = ancho
abline()	Línea definida por una ecuación de una recta	abline(a,b) recta $y=ax+b$ abline(h=) línea horizontal abline(v=) línea vertical abline(lm.obj) recta estimada por una función
title()	Títulos	title(main=..., sub=..., xlab=...,)
axis()	Ejes	axis(n, ..) n=1 horizontal, 2=vertical
text()	Texto	text(x,y,labels) con x, y son las coordenadas; y <i>labels</i> contiene el texto
segments() arrows()	Segmentos Flechas	segments(x1,y1,x2,y2) coordenadas arrows(x1,y1,x2,y2) coordenadas
legend()	Dibuja una leyenda	legend(x,y,label) con x, y coordenadas

Ejemplo: Gráfico

```
> z <- c(73,87,91,99,107,123,122,141,130,144,137,151)
> dev.new()
> plot(x, y, type="l", lwd=3, col="red" )
> lines(x, z, lwd=3, col="blue" )
> legend(17,70, leg=c("Serie Y","Serie Z"), c("red","blue"), cex=0.7)
> abline(h=100)
```



Funciones gráficas de alto nivel

Parámetro	Descripción	Parámetros - Comentarios
<code>plot()</code>	Gráfico de dispersión	
<code>barplot()</code>	Gráfico de barras	Un vector (una barra por elemento) Una matriz (una barra por columna, y cada una de estas barras se divide en barras definidas por las filas)
<code>boxplot()</code>	Gráfico de caja	Un vector o una fórmula (grupos)
<code>hist()</code>	Histograma	Un vector
<code>pie()</code>	Gráfico de sectores	Un vector
<code>coplot()</code>	Explora varias variables	Muestra varios gráficos entre las variables para explorar la relación
<code>pairs()</code>	Gráficos de dispersión entre todas las variables de una matriz	Matriz o <i>dataframe</i>

Salvar los gráficos de R

- Copiar al **portapapeles**: *metafile* o imagen (*bitmap*)
 - El icono copiar, lo hace como *metafile*
- Opción de **menú** para salvar los gráficos (“File” / “Save As”)
 - *metafile, postscripts, pdf, png, bmp, tiff, jpeg (50%, 75%, 100% quality)*
- **Funciones de R** para salvar gráficos con distintos formatos y calidades, según parámetros
- Se utilizan de la siguiente forma:
 - Se **abre un dispositivo** del formato que se deseé
 - ***pdf(), png(), jpeg(), bmp(), tiff(), postscript()***
 - Se usa la función o funciones en R que generan el gráfico o los gráficos
 - Se cierra el dispositivo con la función ***dev.off()***

Ejemplo: Salvar gráficos

```
> ## Salvar gráficos
> ## Formato PDF
> pdf("C://Bioestadistica con R/Temp/Graficos 1.pdf")
> plot(x, y, type="l", lwd=3, col="red" )
> plot(x, z, type="l", lwd=3, col="blue" )
> dev.off()
windows
  2
>
> ## Formato JPG (solo salva el último)
> jpeg("C://Bioestadistica con R/Temp/Graficos 2.jpg" , quality=100)
> plot(x, z, type="l", lwd=3, col="blue" )
> dev.off()
windows
  2
```

Ejercicios

- Leer el fichero de datos: “Bajo peso al nacer.csv”, y realizar **un gráfico de dispersión** con la función *plot()* de las variable “peso” (Y) respecto a “edad” (X), que tenga las siguientes características:
 - *xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")*
 - Un título con un tamaño de fuente un poco menor que lo normal (0.9)
 - Etiquetas en los ejes (“edad”, “peso”) con un tamaño un poco mayor que el normal (1.2)
 - Puntos sólidos (*pch=16*) de color rojo con un tamaño menor que lo normal (0.8)
- Repetir el mismo **gráfico de dispersión** (peso respecto a edad), pero
 - Usar las opciones *ann=FALSE* y *axes=FALSE* (es decir, sin textos ni ejes)
 - Dibujar los **ejes** x, y con la función *axis()* (1=x, 2=y)

Estadística Aplicada a la Investigación Biomédica con R

5 Estadística Descriptiva

- ✓ **Estadística Descriptiva**
- ✓ **Descripción de una variable categórica**
- ✓ **Descripción de una variable continua**
- ✓ **Normalidad**
- ✓ **Funciones de probabilidad**

Estadística descriptiva. Tipos de variables

- **Estadística descriptiva**
 - Resume y describe cada una de las variables del conjunto de datos
 - Proporciona medidas de resumen numéricas y gráficos
- **Variables categóricas. Variables cualitativas o discretas**
 - Número de categorías finitas predeterminadas. Códigos alfabéticos o numéricos
 - Describen una calidad de la muestra
 - **Variables ordinales:** las categorías tienen un orden natural
- **Variables cuantitativas. Variables continuas**
 - Variables medidas en una escala numérica
 - **Rango** definido: mínimo y máximo

Descripción de una variable categórica

- **Medidas resumen**
 - **Frecuencias absolutas**
 - Número de observaciones que presentan cada una de las categorías
 - **Frecuencias relativas**
 - Cociente entre las frecuencias absolutas y el tamaño muestral
 - Se expresa en **proporciones** o **porcentajes**
- **Gráficos**
 - **Diagrama de sectores**
 - Representación gráfica en forma de círculo con áreas proporcionales a la frecuencia absoluta o relativa de cada categoría
 - **Diagrama de barras**
 - Representación gráfica donde en el eje de abscisas se representan las categorías y en el eje de ordenadas las frecuencias relativas o absolutas

Ejemplo: Descripción de una variable categórica

```
> ## Fichero Datos: Bajo peso al nacer
> xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
> dim(xx)
[1] 189 10
> ## Frecuencias absolutas
> t1 <- table(xx$bajo_pes)
> t1
 0   1
130 59

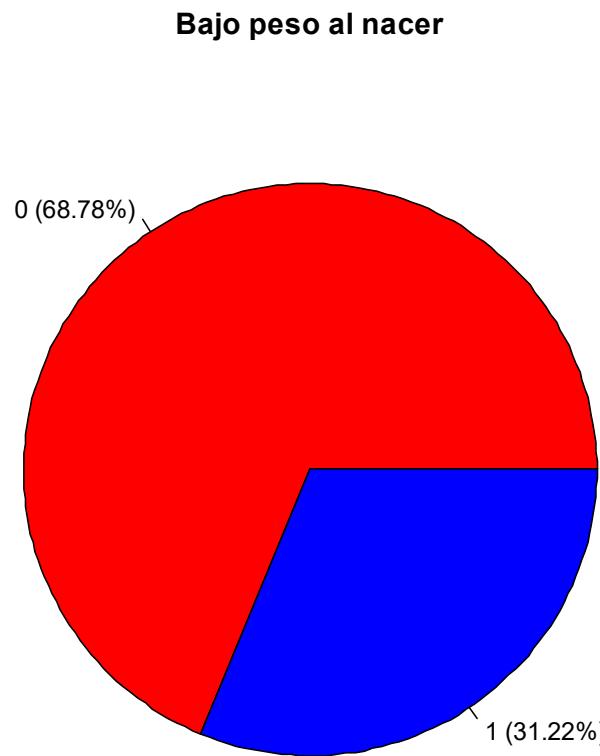
> ## Frecuencias relativas. Proporciones
> p1 <- prop.table(t1)
> p1
 0       1
0.6878307 0.3121693

> ## Porcentajes
> p1 * 100
 0       1
68.78307 31.21693
> round( p1 * 100, dig=2)
 0   1
68.78 31.22
```

- Las funciones básicas son **table()** y **prop.table()** cuyo argumento es una tabla
- Las funciones de gráficos **pie()** y **barplot()** también tienen como argumento una tabla

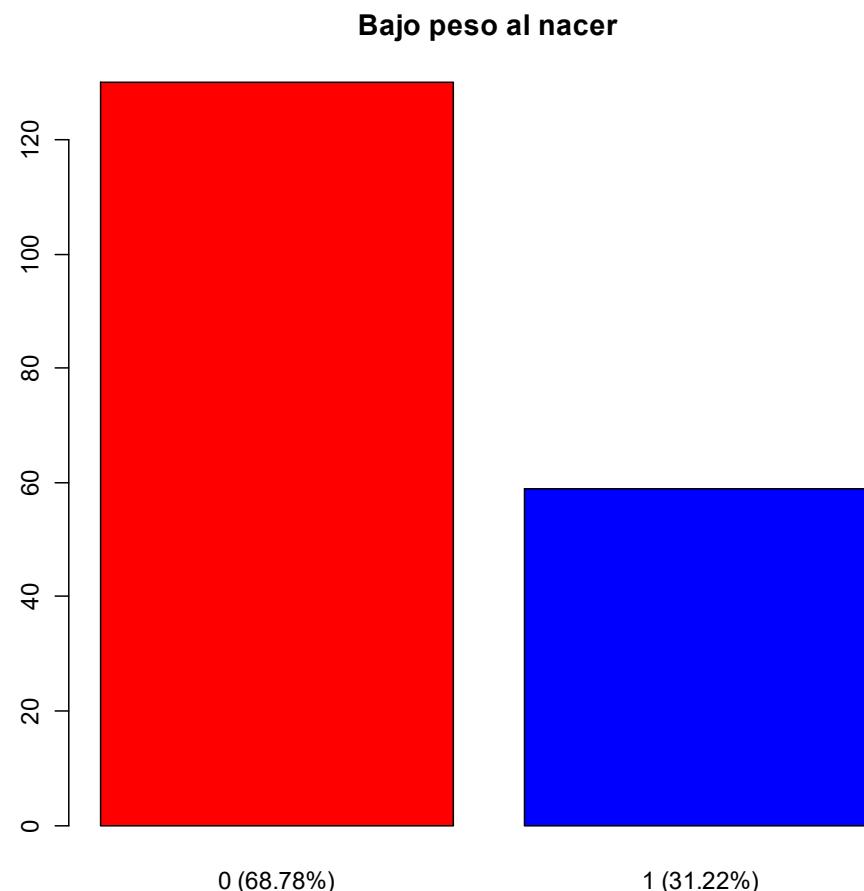
Ejemplo: Descripción de una variable categórica

```
> ## Gráfico de sectores  
> dev.new()  
> pie(t1)  
> pie(t1, col=c("red","blue"), main="Bajo peso al nacer",  
+       labels=paste(names(t1), " (", round( p1 * 100, dig=2), "%)", sep=""))
```



Ejemplo: Descripción de una variable categórica

```
> ## Gráfico de barras  
> dev.new()  
> barplot(t1)  
> barplot(t1, col=c("red","blue"), main="Bajo peso al nacer",  
+           names.arg=paste(names(t1), " (", round( p1 * 100, dig=2), "%)", sep=""))
```



Descripción de una variable continua

- **Medidas resumen de tendencia central**
 - **Media:** es la media aritmética, el centro de gravedad de la muestra
 - **Mediana:** valor central en un conjunto de datos ordenados, deja la mitad de datos a la derecha y la mitad a la izquierda

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

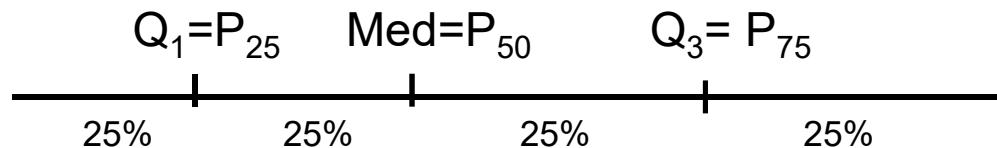
mediana = $x_{\left(\frac{n+1}{2}\right)}$

$x_{(1)}, x_{(2)}, \dots, x_{(n)}$ datos ordenados

- **Media recortada al p% (trimmed):** es la media aritmética de los datos, excluyendo los $(p/2)\%$ más altos y los $(p/2)\%$ más bajos, después de ordenar
 - Se excluyen los **outliers**, valores alejados
- La mediana y la media recortada son **estimadores robustos** porque no son tan sensibles como la media a los valores alejados

Descripción de una variable continua

- **Medidas de posición o localización**
 - El **cuantil o percentil de orden p** de una distribución ($0 < p < 1$): valor P_p tal que una proporción p de valores de los datos $\leq P_p$
 - **Cuartiles:** percentiles 0.25, 0.50, 0.75
 - **Terciles:** percentiles 0.3333, 0.6666
 - **Deciles:** percentiles 0.10, 0.20, ..., 0.90



Descripción de una variable continua

- **Medidas de dispersión**
 - Medidas que cuantifican cómo de dispersos están los datos alrededor de una medida de tendencia central
 - **Varianza:** es la media de las desviaciones al cuadrado respecto a la media
 - **Desviación típica:** es la raíz cuadrada de la varianza, y es una medida de variabilidad en las mismas unidades que los datos de la muestra
 - **Coeficiente de variación:** expresa la variabilidad en una medida sin unidad

$$\text{Var} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$$

$$SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

$$CV = 100 * \frac{SD}{\bar{x}}$$

Descripción de una variable continua

- **Medidas de dispersión**
 - **Rango:** es la diferencia entre el valor máximo y el mínimo
 - **Rango intercuartílico:** es la diferencia entre los cuartiles primero y tercero, y es una medida de la dispersión en el 50% de los datos centrales
 - **MAD (median absolute deviation):** es la mediana de las desviaciones en valor absoluto de las observaciones con respecto a la mediana

$$\text{Rango} = \max_{i=1,\dots,n}(x_i) - \min_{i=1,\dots,n}(x_i)$$

$$\text{MAD} = \text{med}_{i=1,\dots,n} \left\{ \left| x_i - \text{med}_{i=1,\dots,n}(x_i) \right| \right\}$$

$$\text{IQR} = Q_3 - Q_1$$

- El rango intercuartílico y el MAD son **estimadores robustos** porque no son tan sensibles como la desviación típica a los valores alejados, ya que están basados en la mediana y en las medidas de posición

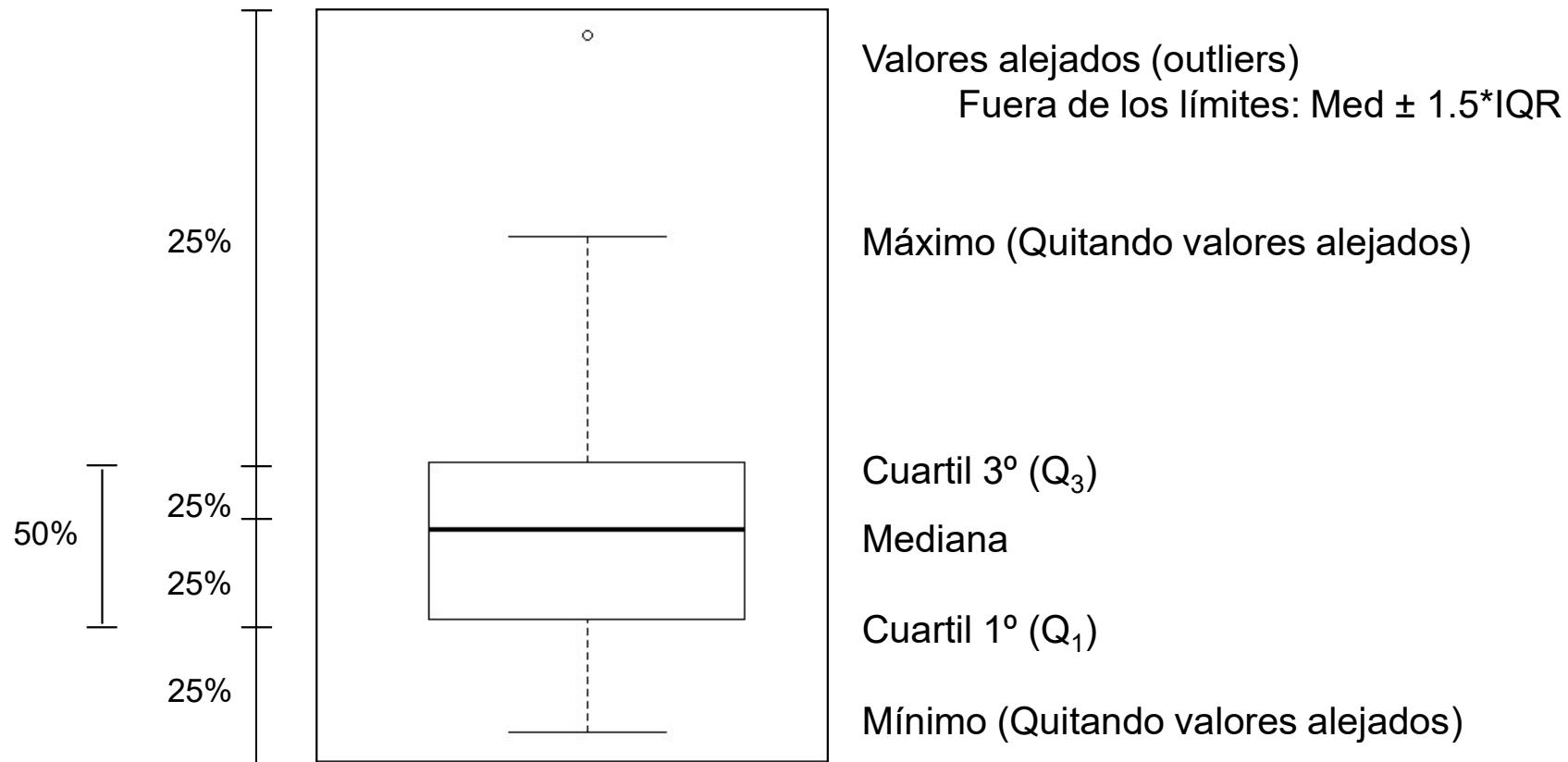
Descripción de una variable continua. Funciones

<i>mean()</i>	Media y media recortada (..., trim=0.10)
<i>median()</i>	Mediana
<i>quantile()</i>	Percentiles (..., c(0.10,))
<i>range()</i>	Rango
<i>var()</i>	Varianza
<i>sd()</i>	Desviación típica
<i>IQR()</i>	Rango intercuartílico
<i>mad()</i>	MAD (median absolute deviation)

- Todas estas funciones tienen un parámetro importante, para tratar con **missings**
 - *nombre.funcion* (.... , **na.rm=TRUE**) que significa “NA remove”
 - Si no se pone esta opción, y el vector contiene algún NA, el resultado de la función es NA

Descripción de una variable continua. Gráficos

- **Histograma**
 - Diagrama de barras donde se representan las frecuencias de la variable agrupada en intervalos
- **Gráfico de caja (Boxplot)**

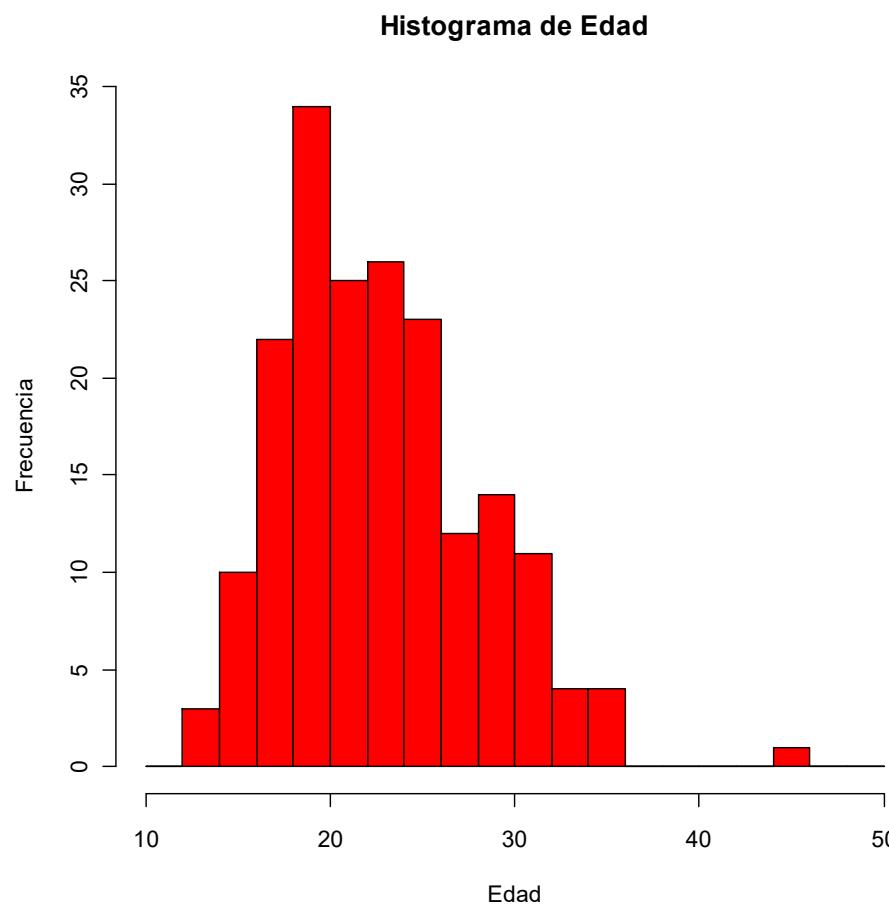


Ejemplo: Descripción de una variable continua

```
> ## Medidas de tendencia central
> mean(xx$edad)
[1] 23.23810
> median(xx$edad)
[1] 23
> mean(xx$edad, trim=0.10)
[1] 22.89542
>
> ## Cuartiles y percentiles
> quantile(xx$edad, c(0.10,0.25,0.50,0.75,0.90))
10% 25% 50% 75% 90%
 17   19   23   26   31
>
> ## Medidas de dispersión
> var(xx$edad)
[1] 28.07599
> sum( (xx$edad - mean(xx$edad))^2 ) / (length(xx$edad)-1)
[1] 28.07599
> sd(xx$edad)
[1] 5.298678
> 100 * abs(sd(xx$edad) / mean(xx$edad)))
[1] 22.80169
> IQR(xx$edad)
[1] 7
> mad(xx$edad) ## MAD con un factor de corrección para que sea comparable con SD
[1] 5.9304
```

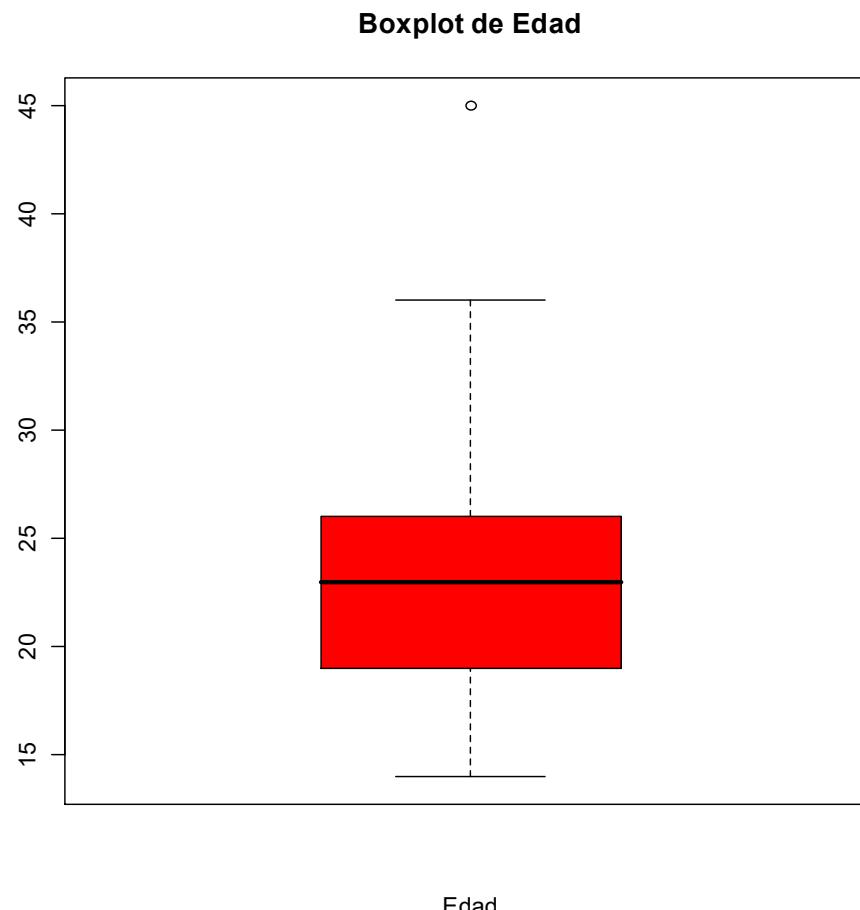
Ejemplo: Descripción de una variable continua

```
> ## Histograma  
> dev.new()  
> hist(xx$edad)  
> hist(xx$edad, col="red", main="Histograma de Edad",  
+       xlab="Edad", ylab="Frecuencia",  
+       xlim=c(10,50), breaks=seq(10,50, by=2))
```



Ejemplo: Descripción de una variable continua

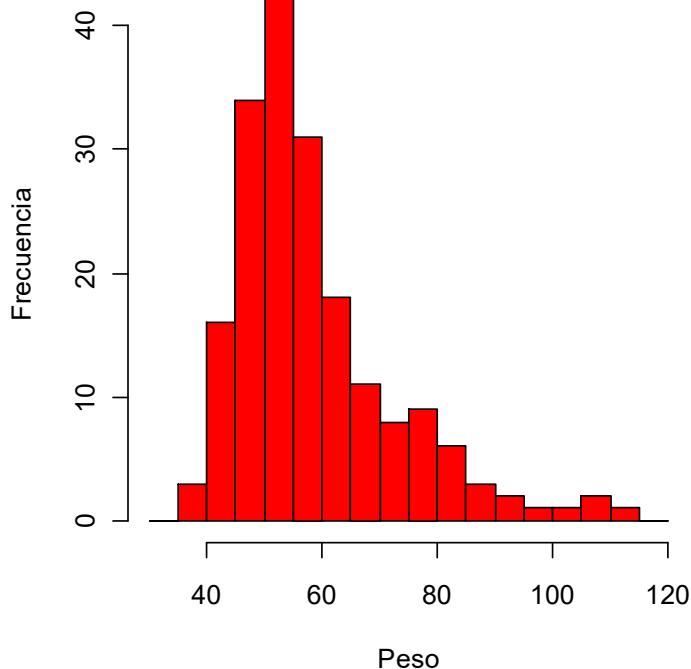
```
> ## Boxplot  
> dev.new()  
> boxplot(xx$edad)  
> boxplot(xx$edad, col="red", main="Boxplot de Edad", xlab="Edad")
```



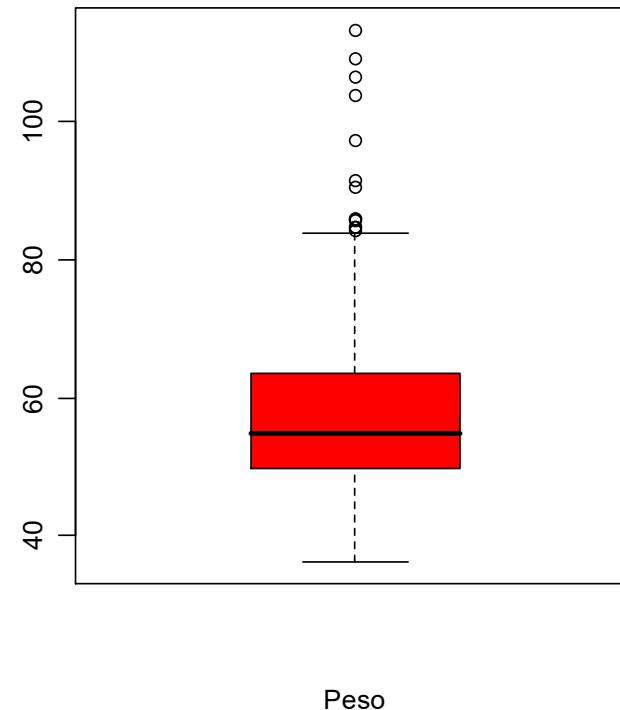
Ejemplo: Descripción de una variable continua

```
> ## Histograma y boxplot de la variable peso
> dev.new()
> par(mfrow=c(1,2))
> hist(xx$peso, col="red", main="Histograma de Peso de la madre",
+       xlab="Peso", ylab="Frecuencia",
+       xlim=c(30,120), breaks=seq(30,120, by=5))
> boxplot(xx$peso, col="red", main="Boxplot de Peso de la madre", xlab="Peso")
```

Histograma de Peso de la madre



Boxplot de Peso de la madre



Ejemplo: función *summary()*

```
> xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv")
> summary(xx)
      id      bajo_pes     edad       peso      raza   fumador part_pre hta
Min.   : 4.0   0:130    Min.   :14.00   Min.   : 36.29  1:96   0:115   0:159   0:177
1st Qu.: 68.0  1: 59    1st Qu.:19.00   1st Qu.: 49.90  2:26   1: 74    1: 24    1: 12
Median :123.0                    Median :23.00    Median : 54.89  3:67   2: 5     3: 1
Mean   :121.1                    Mean   :23.24    Mean   : 58.88
3rd Qu.:176.0                    3rd Qu.:26.00    3rd Qu.: 63.50
Max.   :226.0                    Max.   :45.00    Max.   :113.40

  irr_urin    visi_med
0:161      Min.   :0.0000
1: 28      1st Qu.:0.0000
             Median :0.0000
             Mean   :0.7937
             3rd Qu.:1.0000
             Max.   :6.0000
```

- La función ***summary()*** tiene como argumento un data frame, y muestra:
 - Frecuencias absolutas para variables categóricas
 - Estadísticos básicos para variables continuas
 - Número de missings de cada variable, si hay alguno

Comprobando la normalidad de una variable

- **Normalidad:** algunas pruebas estadísticas requieren que las variables se distribuyan según una distribución normal, y esto debe ser comprobado
- **Comprobando la normalidad. Test de hipótesis**
 - H_0 : los datos se distribuyen según una distribución normal
 - **Test de Shapiro-Wilks**
 - Muestras pequeñas (< 30 , < 50)
 - **Test de Kolmogorov-Smirnov con corrección**
 - Muestras grandes (> 30 , > 50). Es un test más general, y se puede usar para cualquier distribución, además de la normal
- **Comprobando la normalidad. Métodos gráficos**
 - **Q-Q Plot:** se muestran los quantiles observados en los datos frente a los teóricos
 - Si los puntos se ajustan a una recta, los datos siguen la distribución teórica

Ejemplo: Normalidad

```
> library(nortest)
>
> ## Test de hipótesis
> shapiro.test(xx$edad)
    Shapiro-Wilk normality test
data: xx$edad
W = 0.9598, p-value = 3.19e-05

> lillie.test(xx$edad)
    Lilliefors (Kolmogorov-Smirnov) normality test
data: xx$edad
D = 0.0945, p-value = 0.000303

> shapiro.test(xx$peso)
    Shapiro-Wilk normality test
data: xx$peso
W = 0.8933, p-value = 2.242e-10

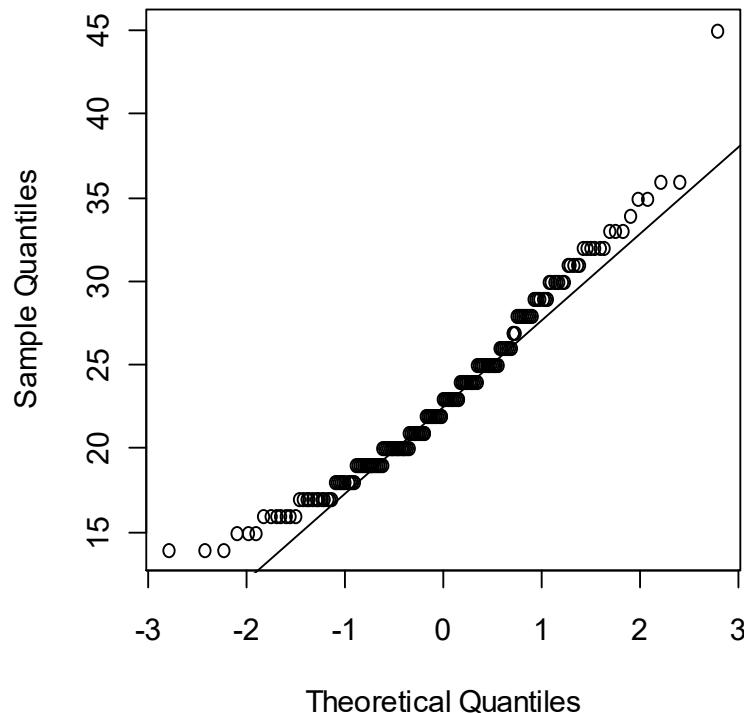
> lillie.test(xx$peso)
    Lilliefors (Kolmogorov-Smirnov) normality test
data: xx$peso
D = 0.1523, p-value = 1.583e-11
```

- El test de Kolmogorov está en la librería **nortest** que hay que cargarla previamente
- Rechazamos la hipótesis de normalidad en las 2 variables

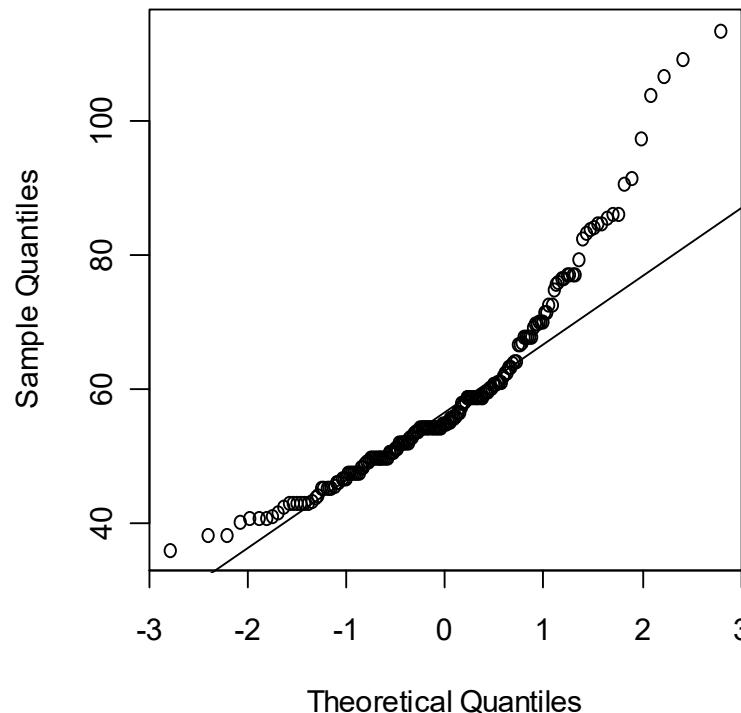
Ejemplo: Q-Q Plot

```
> dev.new()  
> par(mfrow=c(1,2))  
> qqnorm(xx$edad, main="Normal Q-Q Plot - Edad")  
> qqline(xx$edad)  
> qqnorm(xx$peso, main="Normal Q-Q Plot - Peso")  
> qqline(xx$peso)
```

Normal Q-Q Plot - Edad

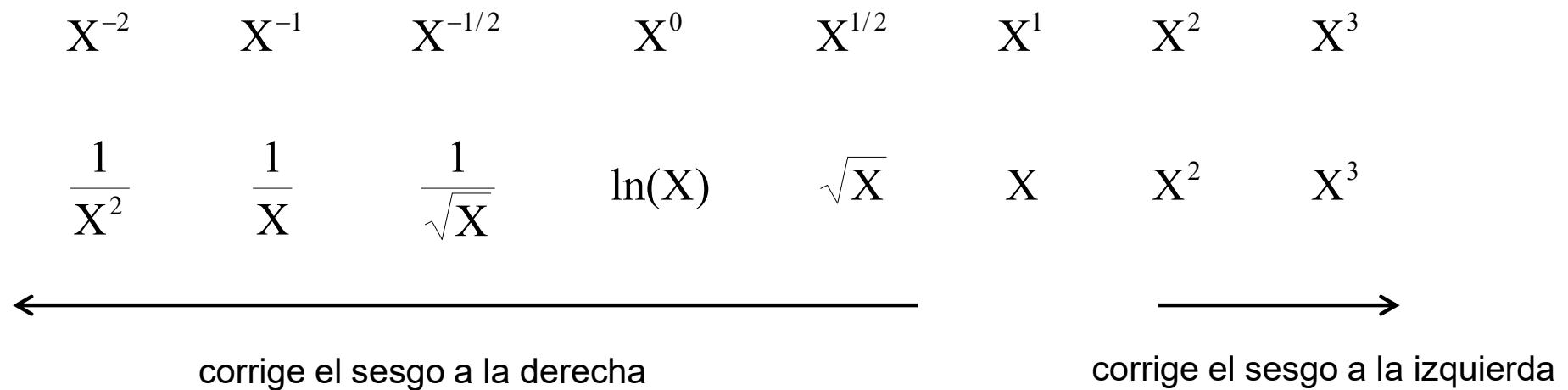


Normal Q-Q Plot - Peso



Transformaciones de una variable

- Se puede transformar una variable continua para conseguir que siga una **distribución normal**, o al menos tenga una **distribución simétrica**
- Se suele usar la familia de **Transformaciones Potencia** X^p con $p \neq 0$
- Resulta útil probar con un conjunto más sencillo e interpretable de estas transformaciones



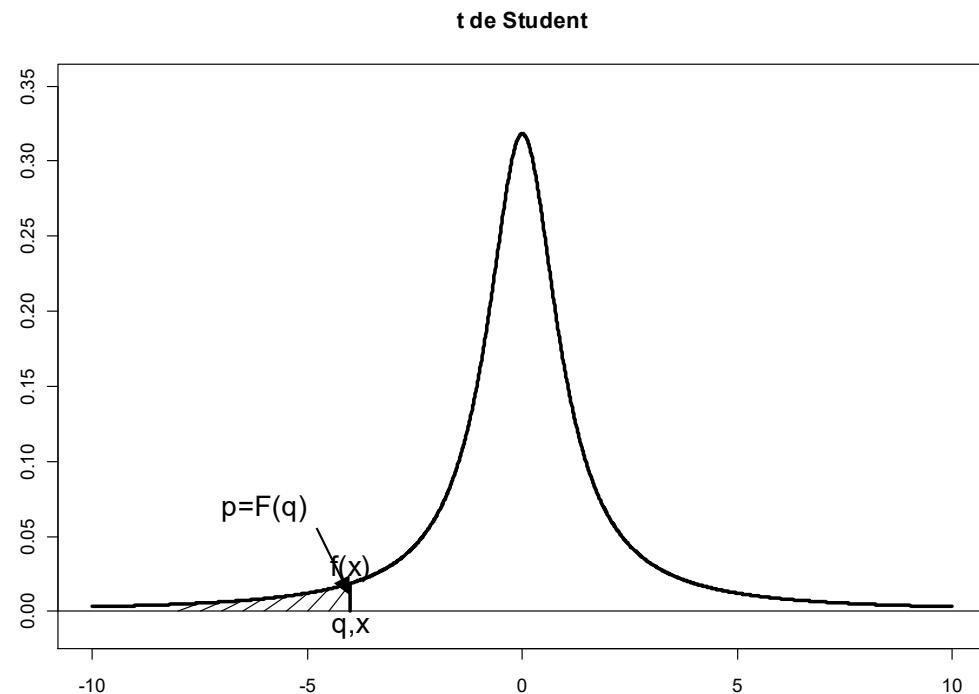
Distribuciones de probabilidad

Distribución	Nombre en R	Parámetros
Beta	beta	shape1, shapw2, ncp
Chi-cuadrado	chisq	df, ncp
Binomial	binom	size, prob
Exponencial	exp	rate
F	f	df1, df2, ncp
Gamma	gamma	shape, scale
Log-normal	Inorm	meanlog, sdlog
Normal	norm	mean, sd
Poisson	pois	lambda
T de student	t	df, ncp
uniforme	unif	min, max

- Otras distribuciones: Cauchy, geométrica, hipergeométrica, logística, binomial negativa, Weibull, Wilcoxon

Distribuciones de probabilidad

Prefijo	Función	Normal	t de Student	Chi-cuadrado
d	Función de densidad $f(x)$	<i>dnorm(x)</i>	<i>dt(x)</i>	<i>dchisq(x)</i>
p	Función de distribución $F(x)$	<i>pnorm(q)</i>	<i>pt(q)</i>	<i>pchisq(q)</i>
q	Función cuantil	<i>qnorm(p)</i>	<i>qt(p)</i>	<i>qchisq(p)</i>
r	Generación aleatoria	<i>rnorm(n)</i>	<i>rt(n)</i>	<i>rchisq(n)</i>



$$F(x) = P(X \leq x)$$

$$q/F(q) = p$$

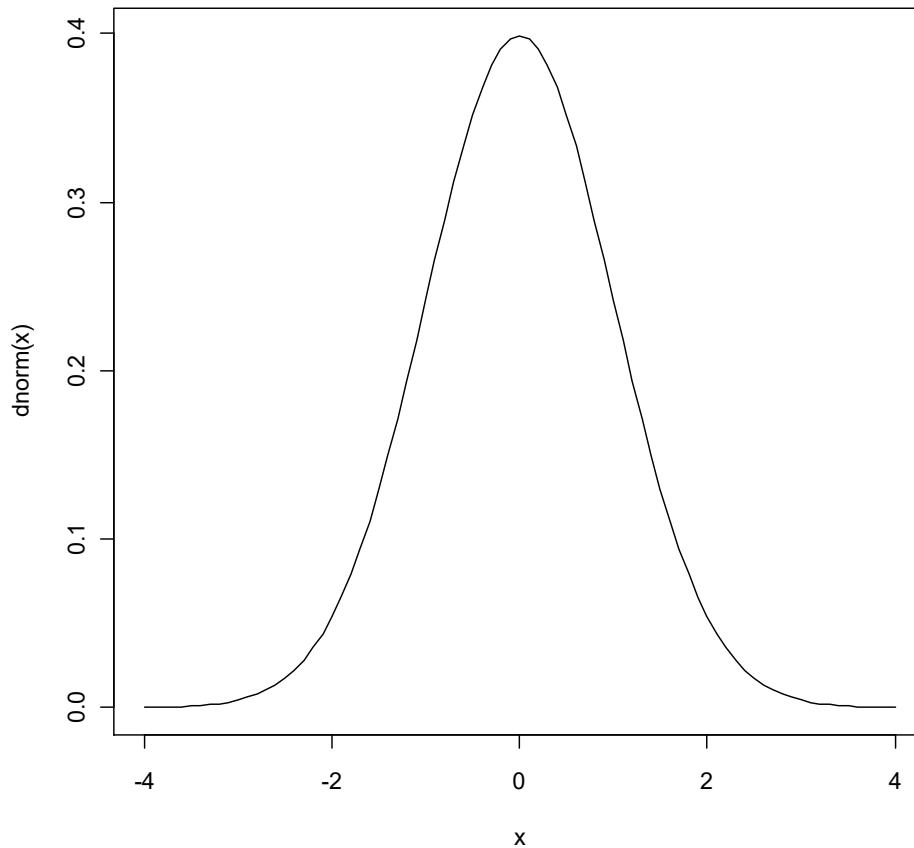
El valor “ q ” deja una probabilidad “ p ” a la izquierda

Ejemplo: Distribuciones de probabilidad

```
> ## Función de densidad de una Normal estándar
> dnorm(0, mean=0, sd=1)
[1] 0.3989423
>
> ## Función de distribución de una Normal estándar
> pnorm(1.96, mean=0, sd=1)
[1] 0.9750021
> pnorm(0, mean=0, sd=1)
[1] 0.5
> pnorm(-1.96, mean=0, sd=1)
[1] 0.02499790
>
> ## Función de cuantiles de una Normal estándar
> qnorm(0.025, mean=0, sd=1)
[1] -1.959964
> qnorm(0.975, mean=0, sd=1)
[1] 1.959964
> qnorm(0.025, mean=0, sd=1, lower.tail=FALSE)
[1] 1.959964
>
> ## Generar Normal estándar
> norm.st <- rnorm(1000, mean=0, sd=1)
> head(norm.st)
[1] 0.24000268 1.43046498 1.03060564 0.05340297 -1.14011081 -0.85075978
> mean(norm.st)
[1] 0.01175251
> sd(norm.st)
[1] 1.048094
```

Ejemplo: Distribuciones de probabilidad

```
> ## Gráfico de una Normal estándar  
> dev.new()  
> x<-seq(-4,4,by=0.1)  
> plot(x,dnorm(x), type="l")  
>
```



Ejercicio

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
1. Describe **las variables categóricas** de este conjunto de datos: IVHX, RACE, TREAT, SITE, DFREE
 - Frecuencias absolutas y proporciones
 - Los 5 gráficos de barras en una sola ventana usando *par(mfrow=c(2,3))*
 2. Describe **las variables cuantitativas** de este conjunto de datos: AGE, BECK, NDRUGTX
 - Media, SD
 - Una ventana, mostrando los 3 histogramas en la 1º fila y los 3 boxplots en la 2º
 - Comprobar la normalidad de las variables con los 2 tests: Shapiro-Wilks y Kolmogorov-Smirnov

Estadística Aplicada a la Investigación Biomédica con R

6 Programación y Funciones en R

- ✓ Programación en R
- ✓ Ejecución condicional con *if*
- ✓ Bucles
- ✓ Funciones en R
- ✓ Funciones de la familia *apply*

Programación en R

- R es un **lenguaje de expresiones**
 - Los comandos son expresiones o funciones, que **retornan un resultado**
- Los comandos pueden ser **agrupados** mediante **llaves** separadas por ;
{ expr_1 ; ; expr_n } y retorna el resultado de la última expresión
- Las expresiones también se pueden agrupar

```
{  
  expr_1  
  ....  
  expr_n  
}
```
- El uso de la agrupación de expresiones y de funciones ayuda a hacer **comprendible** el código de programación

Ejecución condicional: *if*

- Sintaxis: ***if (expr_1) expr_2 [else expr_3]***
 - *expr_1* es una **expresión lógica**, que tiene TRUE/FALSE como resultado
 - Si *expr_1* es TRUE se ejecuta *expr_2*
 - Si *expr_1* es FALSE se ejecuta *expr_3* (opcional)
 - Si se desea ejecutar varias expresiones, se agrupan entre llaves { }
- **Operadores lógicos**
 - Igualdad es **==**, no confundir con =

```
if ( a > 6 ) {
  b = 7
} else {
  b = 3
}
```

<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
==	Igualdad
!=	Desigualdad
&	Y (AND)
 	O (OR)
!	Negación lógica

Ejemplo: Ejecución condicional con *if*

```
> a = 4
> if ( a > 6 ) b = 5 else b = 8
> b
[1] 8
>
> if ( a > 6 ) {
+   b = 7
+ } else {
+   b = 3
+ }
> b
[1] 3
>
> a > 6
[1] FALSE
```

- Se evalúa la condición dentro del paréntesis. Como es FALSE se ejecuta las órdenes que hay entre {} en el else

Bucles: *for*, *repeat*, *while*

- Sintaxis: ***for (var in expr_1) expr_2***
 - *var* es una variable, y *expr_1* suele ser un **rango a:b** o un **conjunto de valores**
 - *expr_2* es un **conjunto de expresiones** que dependen de la variable *var*
- Sintaxis: ***while (condition) expr***
 - La expresión *expr* se ejecuta mientras la **condición sea cierta**
- Sintaxis: ***repeat expr***
 - La expresión *expr* se repite hasta que se ejecute la función ***break***
 - La función ***break*** se puede utilizar para forzar la interrupción de cualquier bucle, y la función ***next*** para pasar a la siguiente iteración del bucle
- R dispone de **cálculo vectorial** que es muy rápido, y que puede ahorrar bucles: suma de vectores, condiciones sobre vectores, ...

Ejemplo: Bucle con *while*

```
> x = c ( 4, 10, 11, 13, 15, 16, 17, 18, 19, 20, 22, 24, 25, 27, 29, 30, 31 )
>
> i <- 1
> while ( x[i] < 20 )
+ {
+   print ( x[i] )
+   i <- i + 1
+ }
[1] 4
[1] 10
[1] 11
[1] 13
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
>
```

- Se muestra los valores del vector menores que 20

Ejemplo: Bucle con *repeat*

```
> x
[1]  4 10 11 13 15 16 17 18 19 20 22 24 25 27 29 30 31
>
> i <- 1
> repeat
+ {
+   if ( x[i] >= 20 )
+   { break }
+   else
+   {
+     print ( x[i] )
+     i <- i + 1
+   }
+ }
[1] 4
[1] 10
[1] 11
[1] 13
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
```

- Se muestra los valores del vector menores que 20

Ejemplo: Bucle con *for*

```
> x = c ( 4, 10, 11, 13, 15, 16, 17, 18, 19, 20, 22, 24, 25, 27, 29, 30, 31 )
> for ( i in 1:length(x) )
+ {
+   if ( x [i] >= 20 )
+     { break }
+   else
+     { print ( x[i] ) }
+ }
[1] 4
[1] 10
[1] 11
[1] 13
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
> for ( i in 1:length(x) )
+ { if ( x [i] < 15 ) print ( x[i] ** 2 ) }
[1] 16
[1] 100
[1] 121
[1] 169
> ## Cálculo vectorial
> x[ x < 15 ] ** 2
[1] 16 100 121 169
```

- El cálculo vectorial puede sustituir a algunos bucles *for* y es más rápido

Ejemplo: Bucle con *for*

```
> ## Fichero Datos: Bajo peso al nacer
> xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
> ## Nombres de las variables
> names(xx)
[1] "id"  "bajo_pes" "edad" "peso" "raza"  "fumador" "part_pre" "hta"   "irr_urin" "visi_med"
>
> ## Variables categóricas
> xx$bajo_pes <- factor(xx$bajo_pes)
> xx$raza     <- factor(xx$raza)
> xx$fumador  <- factor(xx$fumador)
> xx$part_pre <- factor(xx$part_pre)
> xx$hta      <- factor(xx$hta)
> xx$irr_urin <- factor(xx$irr_urin)
>
> ## Detectamos las variables que son categóricas
>
> for ( i in 3:10)
+ {
+   if ( is.factor( xx[ , i ] ) == TRUE  )
+     print ("es factor") else print ("no es factor")
+ }
[1] "no es factor"
[1] "no es factor"
[1] "es factor"
[1] "no es factor"
```

Ejemplo: Bucle con *for*

```
> ## Análisis Descriptivo de las variables independientes
> ## - Si X es un factor, mostramos la tabla
> ## - SI X es una variable continua, mostramos la media y la SD
>
> for ( i in 3:10 )
+ {
+   print ( names(xx)[i] )
+   if ( is.factor( xx[, i ] ) == TRUE  )
+   {
+     print ( table (xx[, i ]) )
+   }
+   else
+   {
+     print ( paste ( "media(SD) =", round( mean(xx[ , i]), dig=2) ,
+                   "(, round( sd(xx[ ,i]),dig=2 ), ")" ))
+   }
+ }
[1] "edad"
[1] "media(SD) = 23.24 ( 5.3 )"
[1] "peso"
[1] "media(SD) = 58.88 ( 13.87 )"
[1] "raza"

  1  2  3
96 26 67
[1] "fumador"

  . . . . . . .
```

Funciones en R

- Una **función** es un conjunto de **instrucciones** de R que están **agrupadas** bajo un **nombre** y que usa **parámetros**
 - Las asignaciones que se hacen dentro de la función son temporales
- **Ventajas** de usar funciones
 - Versatilidad y flexibilidad
 - No repetir código. Agrupación de funciones rutinarias
 - Legibilidad del código

Funciones en R

- **Sintaxis de las funciones**

```
nombre <- function (arg_1, arg_2, .... , arg_p )  
{  
    expresiones  
}
```

- **Llamadas a la función:** *nombre (expr_1, expr_2, ... expr_p)*

- *nombre (arg_2 = expr_2 , arg_4 = expr_4)*

- El valor de la última expresión es el valor que **retorna la función**

- Si se desea retornar varios objetos, se agrupan en una **lista**

- Si se desea **interrumpir** la ejecución de la función se usa **stop()**
- Se suele usar para controlar los parámetros al comienzo de la función y **retornar errores**

Ejemplo: Función

```
> ## Función que calcula las medidas resumen de una variable continua
> AnalisisContinua <- function ( var )
+ {
+   if ( is.factor(var) == TRUE )
+     { stop ( "La variable no es continua" ) }
+
+   ## Medidas de tendencia central
+   mean = mean (var, na.rm=TRUE)
+   median = median(var na.rm=TRUE)
+   mean.trim.10 = mean(var, trim=0.10)
+
+   ## Medidas de Dispersion - Variabilidad
+   sd = sd (var, na.rm=TRUE)
+   iqr = IQR (var, na.rm=TRUE)
+   mad = mad (var, na.rm=TRUE)
+
+   list ( mean=mean, median=median, mean.trim.10 = mean.trim.10,
+         sd=sd, iqr=iqr, mad=mad)
+ }
```

- AnalisisContinua es el **nombre de la función**. Las expresiones se agrupan entre { }
- **var** es el único **parámetro**, variable continua para analizar
- Si la variable es un factor (categórica) se detiene la ejecución de la función con **stop**
- El **cuerpo de la función** es el cálculo de las medidas resumen
- Los **resultados** se devuelven en una **lista** (nombre del elemento = variable interna)

Ejemplo: Llamada a la función

```
> summ.edad <- AnalisisContinua (xx$edad)
> summ.raza <- AnalisisContinua (xx$raza)
Error in AnalisisContinua(xx$raza) : La variable no es continua
> summ.peso <- AnalisisContinua (xx$peso)
> summ.peso
$mean
[1] 58.88361

$median
[1] 23

$mean.trim.10
[1] 22.89542

$sd
[1] 13.87072

$iqr
[1] 13.60791

$mad
[1] 9.415041
> names(summ.peso)
[1] "mean"          "median"        "mean.trim.10" "sd"           "iqr"          "mad"
> summ.peso$mean
[1] 58.88361
> summ.peso$sd
[1] 13.87072
```

Funciones en R

- **El argumento puntos suspensivos: “...”**
 - Es un argumento especial que permite pasar parámetros de una función a otra
 - Se suele utilizar bastante con *plot()*

```
myfunc <- function (x, y, ... )  
{  
  ....  
  plot(x,y, ...)  
  ....  
}  
myfunc ( var1, var2, main="Titulo", cex=0.8)
```

- Si dentro de una función aparece una variable que no ha sido definida ni es un parámetro, la función la busca fuera de la función (en el *worspace()*)
 - No es una práctica aconsejable

R es un lenguaje orientado a objetos

- **R es un lenguaje orientado a objetos:** almacena **los resultados de las funciones en objetos** que pueden ser mostrados, analizados o utilizados en programación como se deseé
 - Los objetos se pueden salvar en un fichero **RData** para usarlos posteriormente
- **Clase de los objetos, *class()***
 - Se pueden crear nuevas clases
 - Los **resultados de los análisis** de las funciones importantes de un paquete, suelen ser **objetos** de clases definidas por el creador de la función
- **Funciones genéricas** que son distintas según la clase del objeto que se les pase como parámetro
 - ***print()*, *summary()*, *plot()***

Ejemplo: Funciones de R y Objetos

```
> x1 = c( 12,34,23,34,73,18,36,23,55,62,38,76 )
> x2 = c( 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0 )
> y = c( 16,43,22,45,66,32,47,12,27,72,45,33 )
>
> ## Regresión Lineal
> out = lm ( y ~ x1 + x2 )
> out

Coefficients:
(Intercept)          x1            x2
16.1230        0.4837       5.3999

> class(out)
[1] "lm"
> names(out)
[1] "coefficients"   "residuals"      "effects"        "rank"           "fitted.values"
[6] "assign"          "qr"             "df.residual"    "xlevels"        "call"
[11] "terms"          "model"
> out$coeff
(Intercept)          x1            x2
16.1230461  0.4837273  5.3999094
> out$residuals
     1         2         3         4         5         6         7         8
-5.927773 10.430227 -5.248773  7.030318  9.164955  7.169863 13.462773 -20.648682
     9        10        11        12
-21.127954 20.485955  5.095409 -19.886317
```

Ejemplo: Funciones y Objetos

```
> summary(out)
Residuals:
    Min      1Q  Median      3Q     Max 
-21.128 -9.417   6.063   9.481  20.486 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 16.1230    10.1995   1.581   0.1484    
x1          0.4837     0.2381   2.032   0.0727 .  
x2          5.3999    9.7482   0.554   0.5931    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.82 on 9 degrees of freedom
Multiple R-squared:  0.3981,    Adjusted R-squared:  0.2644 
F-statistic: 2.977 on 2 and 9 DF,  p-value: 0.1018

> summary(out)$coeff
            Estimate Std. Error   t value Pr(>|t|)    
(Intercept) 16.1230461 10.1995420 1.5807618 0.1483906  
x1          0.4837273  0.2380531  2.0320138 0.0726975  
x2          5.3999094  9.7482363  0.5539371 0.5931159  
> summary(out)$coeff[, 4]  ## P-values
(Intercept)           x1           x2
0.1483906 0.0726975 0.5931159
```

- La función **summary()** es una función genética, lo que significa que cuando la usamos con un objeto **out** de clase “**Im**”, realmente ejecuta una función llamada **summary.Im()**

Funciones de la familia *apply*

- **Función *apply***
 - Permite **aplicar una función** sobre las filas o columnas de **una matriz**
 - ***apply(X, MARGIN, FUN, ...)***
 - X es la matriz, **MARGIN** indica la dimensión donde se aplica la función, tomando los valores 1 para filas y 2 para columnas
 - **FUN** es una **función** de R o creada por el usuario
- **Función *tapply***
 - Permite aplicar una función sobre **grupos del array X**, para los grupos definidos por un **factor INDEX**
 - ***tapply(X, INDEX, FUN, ...)***
- **Función *lapply***
 - Permite aplicar una función sobre los componentes de **una lista**
 - ***lapply(X, FUN, ...)***

Ejemplo: *apply*

```
> ## apply
> x <- matrix(1:10, 2, 5)
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> apply(x,1,mean)
[1] 5 6
>
> length(apply(x,1,mean))
[1] 2
> is.vector(apply(x,1,mean))
[1] TRUE
> apply(x,1,mean)[1]
[1] 5
> apply(x,1,mean)[2]
[1] 6
>
```

Ejemplo: *tapply* y *lapply*

```
> ## tapply
> a <- factor( c(1,2,1,1,1,1,2,2,1,1))
> a
[1] 1 2 1 1 1 1 2 2 1 1
Levels: 1 2
> x <- c (2.3, 4.4, 5.2, 3.3, 2.1, 3, 2, 3.3, 4.1, 2.9)
> tapply(x, a, mean)
      1         2
3.271429 3.233333
>
> ## tapply con 2 vectores definiendo los niveles
> b <- c("H","H","H","H","H","M","M","M","M")
> tapply(x, list(a, b) , mean)
      H         M
1 3.225 3.333333
2 4.400 2.650000
>
> ## lapply
> list3 <- list( a=1:5, b=c(3,NA,2) , c=c(FALSE,TRUE,FALSE,TRUE,TRUE) )
> lapply(list3, mean)
$a
[1] 3

$b
[1] NA

$c
[1] 0.6
```

Ejercicio

- Crear una **función** para analizar las variables categóricas:
 - Nombre de la función: **AnalisisCategorica**
 - La variable como único **parámetro**
 - Obs.: No hace falta controlar si la variable es factor. Algunas variables categóricas no se definen como factores (binarias, ordinales)
 - Como **resultado** de la función se muestra la tabla (función *table()*), las proporciones (función *prop.table()*) y los porcentajes con 2 decimales
- **Ejecutar la función** para algunas de las variables categóricas del fichero de “Bajo peso al nacer.csv” (fumador, raza, hta, ...)

Estadística Aplicada a la Investigación Biomédica con R

7 Análisis de Tablas de Contingencia

- ✓ **Técnicas Estadísticas**
- ✓ **Inferencia Estadística. Contraste de Hipótesis**
- ✓ **Descripción de una tabla de contingencia**
- ✓ **Tests para tablas de contingencia**
- ✓ **Riesgo Relativo y Odds Ratio**

Técnicas estadísticas. Introducción

- **Análisis Descriptivo**
 - Resume la información contenida en el conjunto de los datos
- **Técnicas de Inferencia Estadística**
 - Estima **parámetros poblacionales** a partir de la información de la muestra, acompañados de **intervalos de confianza** que dan la **precisión** de la estimación puntual
- **Contrastes de Hipótesis**
 - Contrastar hipótesis, **comparando** el comportamiento de dos o más grupos o analizando la relación entre dos variables
- **Modelización**
 - Elabora **modelos o funciones** que permiten explicar unas variables a partir de otras, y permite realizar **predicciones**

Inferencia Estadística

- El **objetivo de la Inferencia Estadística** es extraer conclusiones válidas para una población basadas en la información que proporcionan las **muestras**
 - Se pretende realizar estimaciones e inferencias a partir de las muestras extraídas de poblaciones, teniendo en cuenta **la influencia del azar**
 - Los **estadísticos muestrales** se usan para estimar **parámetros poblacionales**, que son desconocidos
- El **error estándar** de un estimador no cuantifica la variabilidad de las observaciones, sino que cuantifica **la precisión** con que el estimador permite estimar el verdadero parámetro
 - Si se toman diferentes muestras de una población, cada vez se obtiene un estimador distinto del parámetro

Inferencia Estadística

- Los **intervalos de confianza** controlan la incertidumbre y la imprecisión de la estimación puntual de los parámetros
 - Un intervalo de confianza al 95% significa que si se toman 100 muestras de una población y se construyen 100 intervalos de confianza para un parámetro, esperamos que **95 de esos intervalos de confianza contengan el parámetro**
 - El 95% expresa el **margin de confianza** del intervalo
 - Los IC99% son más anchos que los IC95%, y los IC90% son más estrechos
- **Ejemplos:** intervalos de confianza de una proporción y de la media
 - Teniendo en cuenta el **error estándar de la media (SEM)**

$$\bar{x} \pm z_{1-\frac{\alpha}{2}} \cdot \frac{\hat{\sigma}}{\sqrt{n}}$$

$$SEM = \frac{\hat{\sigma}}{\sqrt{n}}$$

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Contraste de Hipótesis

- El **objetivo de un Contraste de Hipótesis** es decidir si una determinada hipótesis o afirmación sobre la distribución poblacional es confirmada o invalidada estadísticamente a partir de las observaciones de una muestra
- Llamamos:
 - **H_0 a la hipótesis nula**, la que queremos validar
 - **H_1 a la hipótesis alternativa**, que suele ser la contraria de H_0
- Los contrastes de hipótesis se enfocan a que debe haber **una gran evidencia para invalidar la hipótesis nula H_0**
 - Por tanto, aceptamos H_0 en el sentido que las observaciones no han aportado suficiente evidencia para descartarla

Contraste de Hipótesis

- Los contrastes se basan en una determinada función de los datos de la muestra, llamada **estadístico de contraste**
 - Se basan en la **distribución teórica** del estadístico de contraste suponiendo que la **hipótesis nula H_0 es cierta**
 - Con los datos de la muestra se calcula el **estadístico de contraste muestral** y se **compara** con el valor que esperamos de la distribución teórica si H_0 fuese cierta
- Llamamos **p-valor** de un test a la probabilidad de obtener, suponiendo que H_0 sea cierta, un resultado al menos tan extremo como el que realmente se ha obtenido en la muestra
 - **Valores pequeño del p-valor permiten rechazar H_0** ya que encontramos poca evidencia de que ese resultado muestral se haya podido obtener siendo H_0 cierta

Contraste de Hipótesis

- En un contraste de hipótesis, se pueden adoptar las siguientes decisiones

	H_0 cierta	H_0 falsa
Rechazar H_0	Error de tipo I = α	Correcto
Aceptar H_0	Correcto	Error de tipo II = β

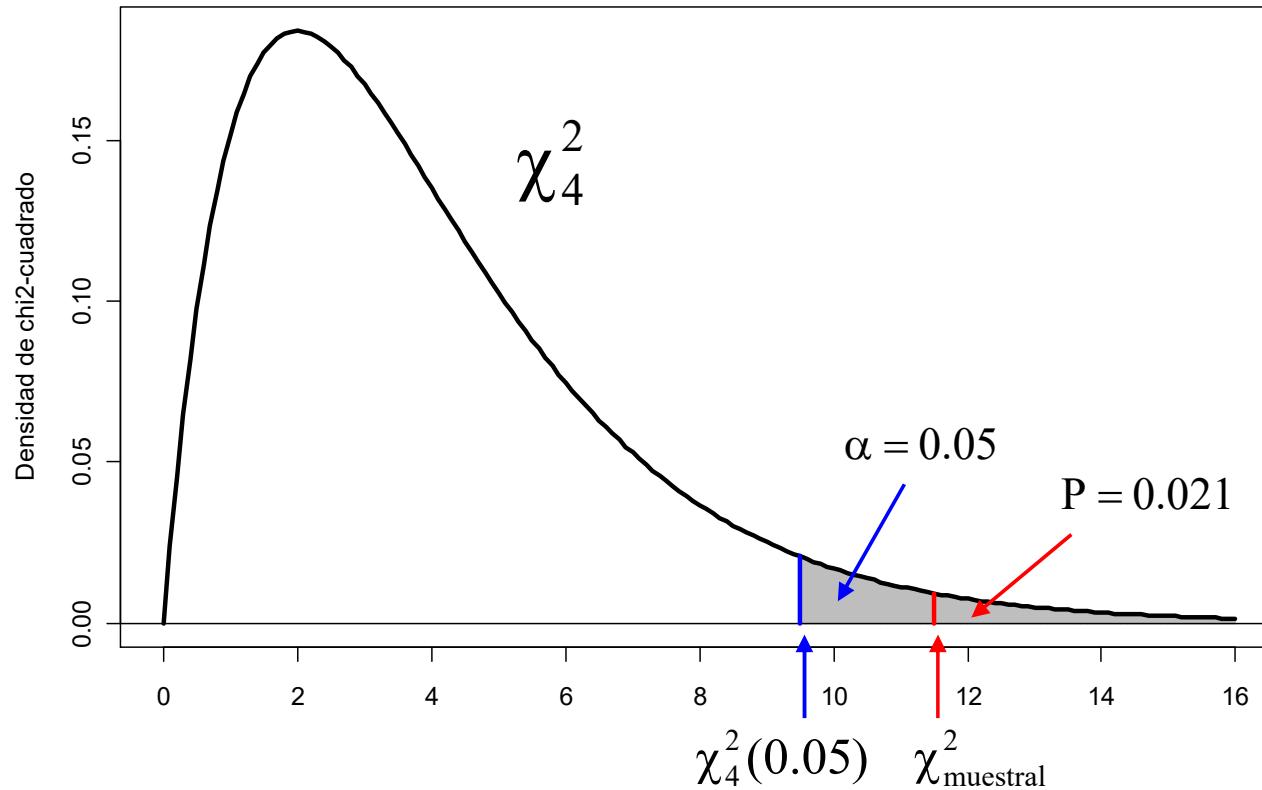
- No se pueden controlar los dos errores a la vez, por eso se suele prefijar el valor del error de tipo I que estamos dispuestos a asumir, y se le llama **nivel de significación** (se suelen tomar valores 0.01, 0.05, 0.1)

$$\alpha = P(\text{ Rechazar } H_0 / H_0 \text{ es cierta})$$

- Llamamos **potencia de un test** al complementario del error de tipo II

$$1 - \beta = P(\text{ Rechazar } H_0 / H_0 \text{ es falsa}) = P(\text{ Aceptar } H_1 / H_1 \text{ es cierta})$$

Contraste de Hipótesis



- Rechazamos H_0 ya que:

$$\chi^2_4(0.05) = 9.49 < 11.5 = \chi^2_{\text{muestral}}$$

$$\alpha = 0.05 > 0.021 = P - \text{valor}$$

- El estadístico de contraste obtenido en la muestra tiene una probabilidad muy pequeña de proceder de la distribución teórica del estadístico si H_0 fuera cierta, y por tanto, rechazamos H_0

Tablas de contingencia

- Las **tablas de contingencia** se usan para explorar la relación entre **dos variables categóricas**
 - Normalmente, se desea explorar la distribución de una variable categórica en los diferentes grupos, definidos por la otra variable
- Para 2 variables categóricas X e Y, con r y c categorías respectivamente, se puede representar **la tabla de contingencia** de la siguiente forma:

		Y				
		1	2	...	c	
X		1	O_{11}	O_{12}	\dots	O_{1c}
		2	O_{21}	O_{22}	\dots	O_{2c}
		
		r	O_{r1}	O_{r2}	\dots	O_{rc}
		$n_{.1}$		$n_{.2}$	$n_{.c}$	

- O_{ij} **número observado** de individuos en la categoría i de X y en la categoría j de Y (**frecuencias**)
- $n_{i.}$ número observado de individuos en la categoría i de X (**distribución marginal de X**)
- $n_{.j}$ número observado de individuos en la categoría j de Y (**distribución marginal de Y**)

Tablas de contingencia

- **Descripción de la tabla de contingencia**
 - Frecuencias absolutas
 - Frecuencias marginales por filas y/o columnas
 - Proporciones por filas y/o columnas
 - Porcentajes por filas y/o columnas
- **Gráfico para una tabla de contingencia**
 - Gráficos de barra
 - Gráficos de sectores

Ejemplo: Descripción de una tabla de contingencia

```
> t1 <- table(xx$raza, xx$bajo_pes )
> t1

      0   1
 1 73 23
 2 15 11
 3 42 25
>
> margin.table (t1, 1) ## Frecuencias marginales por filas

 1  2  3
96 26 67
> margin.table (t1, 2) ## Frecuencias marginales por columnas

 0   1
130 59
>
> prop.table(t1)      ## Proporciones totales

      0          1
 1 0.38624339 0.12169312
 2 0.07936508 0.05820106
 3 0.22222222 0.13227513
```

Ejemplo: Descripción de una tabla de contingencia

```
> t1 <- table(xx$raza, xx$bajo_pes )
> t1

      0   1
1 73 23
2 15 11
3 42 25

> prop.table(t1, 1)    ## Proporciones por filas

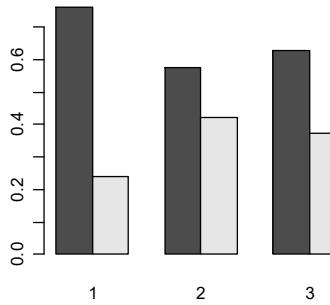
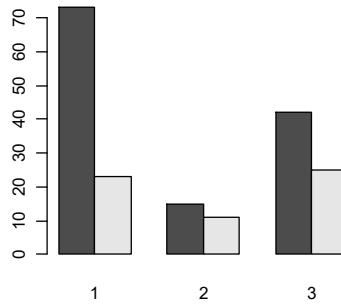
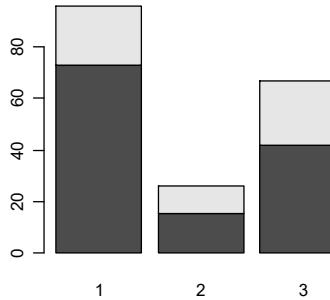
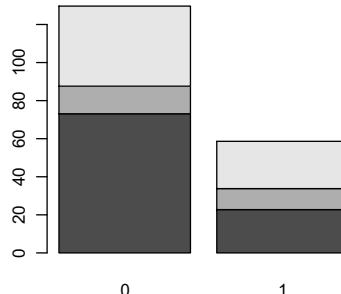
      0       1
1 0.7604167 0.2395833
2 0.5769231 0.4230769
3 0.6268657 0.3731343
> prop.table(t1, 2)    ## Proporciones por columnas

      0       1
1 0.5615385 0.3898305
2 0.1153846 0.1864407
3 0.3230769 0.4237288
> round( 100*prop.table(t1, 1), dig=2 )    ## Porcentajes por filas

      0     1
1 76.04 23.96
2 57.69 42.31
3 62.69 37.31
```

Ejemplo: Gráficos para una tabla de contingencia

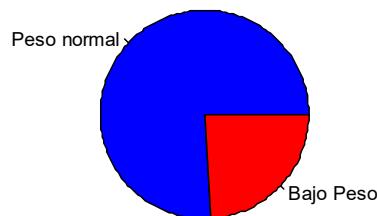
```
> dev.new()  
> par(mfrow=c(2,2))  
> barplot( t1 )  
> barplot( t(t1) )  
> barplot( t(t1), beside=T )  
> barplot( prop.table ( t(t1),2 ), beside=T )
```



Ejemplo: Gráficos para una tabla de contingencia

```
> dev.new()
> par(mfrow=c(2,2))
> lab.bajo=c("Peso normal", "Bajo Peso")
> col.1 = c("blue", "red")
> pie( t1[1,], col=col.1, main="Raza Blanca", lab=lab.bajo )
> pie( t1[2,], col=col.1, main="Raza Negra", lab=lab.bajo )
> pie( t1[3,], col=col.1, main="Otras Razas", lab=lab.bajo )
```

Raza Blanca



Raza Negra



Otras Razas



Test de independencia de chi-cuadrado

- **Test de independencia** entre las dos variables categóricas
 - H_0 : X, Y son independientes
 - H_1 : X, Y no son independientes
 - Se desea contrastar si **la distribución** de una de las variables categóricas es **similar** en todas las categorías de la otra variable
- Bajo la hipótesis nula, los **valores estimados esperados** E_{ij} para la celda ij de la tabla:

$$E_{ij} = \frac{n_{i\cdot} \times n_{\cdot j}}{N}$$

- El **estadístico del test** para evaluar la independencia es:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

que bajo la hipótesis nula de independencia, se distribuye según una **chi-cuadrado** con $(r-1) \times (c-1)$ gl

Test de independencia de chi-cuadrado

- **Limitaciones** del test de chi-cuadrado
 - El test de chi-cuadrado es una test asintótico, para **muestras grandes**
 - Es válido cuando las **frecuencias esperadas** no sean muy pequeñas: se suele admitir hasta un **20-25% de las celdas** con valores esperados **menores que 5**
 - Si no se cumple esta condición, se deben **unir o eliminar categorías** de una de las variables categóricas

Test exacto de Fischer

- El **test exacto de Fisher** es la alternativa al test de chi-cuadrado para **muestras pequeñas** o con celdas con valores esperados bajos
 - **Tablas 2x2**, pero se puede extender para cualquier **Tabla rxc**
 - Se basa en la evaluación de todas las tablas 2x2 que se pueden formar **fijando las frecuencias marginales**
 - La **probabilidad** de observar un conjunto concreto de frecuencias (a,b,c,d) bajo la hipótesis de independencia viene dado por la **distribución hipergeométrica**
 - El valor p del test de Fisher se obtiene sumando las probabilidades de todas las tablas que **muestran diferencias mayores o iguales** que la observada

		Y		$a+b$	$c+d$	n
		1	2			
X	1	a	b			
	2	c	d			
		$a+c$	$b+d$			

$$P = \frac{(a+b)!(a+c)!(c+d)!(b+d)!}{a!b!c!d!N!}$$

Test exacto de Fischer. Ejemplo

- Supongamos que hemos observado la siguiente tabla:

		Y		
		1	11	12
		7	5	12
	X	8	16	

- Fijando las frecuencias marginales, existen 9 posibles tablas:

Tabla 1

		Y		
		0	12	12
		8	4	12
	X	8	16	

Tabla 4

		Y		
		3	9	12
		5	7	12
	X	8	16	

Tabla 7

		Y		
		6	6	12
		2	10	12
	X	8	16	

Tabla 2

		Y		
		1	11	12
		7	5	12
	X	8	16	

Tabla 5

		Y		
		4	8	12
		4	8	12
	X	8	16	

Tabla 8

		Y		
		7	5	12
		1	11	12
	X	8	16	

Tabla 3

		Y		
		2	10	12
		6	6	12
	X	8	16	

Tabla 6

		Y		
		5	7	12
		3	9	12
	X	8	16	

Tabla 9

		Y		
		8	4	12
		0	12	12
	X	8	16	

Test exacto de Fischer. Ejemplo

- Calculamos la probabilidad de cada una de las tablas:

Probabilidad	
Tabla 1	0.00067
Tabla 2	0.01292
Tabla 3	0.08292
Tabla 4	0.23691
Tabla 5	0.33315
Tabla 6	0.23691
Tabla 7	0.08292
Tabla 8	0.01292
Tabla 9	0.00067
TOTAL	1

$$P = \frac{(a+b)!(a+c)!(c+d)!(b+d)!}{a!b!c!d!N!}$$

- El **P-valor del test de Fisher** es la suma de las probabilidades de las tablas que tienen menor o igual probabilidad que la tabla observada:
- $$P - \text{valor} = 0.00067 + 0.01292 + 0.01292 + 0.00067 = 0.02718$$
- En este ejemplo, concluimos que las variables no son independientes ($P=0.027$)

Ejemplo: Tests de independencia

```
> ## Test de chi-cuadrado
> chisq.test(t1)
  Pearson's Chi-squared test
data: t1
X-squared = 5.0048, df = 2, p-value = 0.08189

> chisq.test(t1)$p.value
[1] 0.0818877
> chisq.test(t1)$expected
      0         1
1 66.03175 29.968254
2 17.88360  8.116402
3 46.08466 20.915344
> ## Test exacto de Fisher
> fisher.test(t1)
  Fisher's Exact Test for Count Data
data: t1
p-value = 0.07889
alternative hypothesis: two.sided

> fisher.test(t1)$p.value
[1] 0.07888813
```

- El test chi-cuadrado suele ser **más conservador** que el test exacto de Fisher

Riesgo Relativo

- Esta medida se utiliza para evaluar la **asociación entre la exposición a un factor de riesgo y un evento** (una enfermedad), siendo ambas **variables binarias**

		Enfermedad		Total
Factor	E=1	E=0		
	F=1	a	b	a + b
F=0	c	d		c + d
Total	a + c		b + d	

- El **Riesgo Relativo** es el cociente entre el riesgo de que se produzca la enfermedad cuando está presente el factor y cuando está ausente el factor
 - El riesgo relativo expresa **cuánto más probable** es desarrollar la enfermedad en presencia del factor respecto a la ausencia del factor

$$RR = \frac{P_{F=1}(E=1)}{P_{F=0}(E=1)} = \frac{a/(a+b)}{c/(c+d)}$$

Odds Ratio

- Un **Odd (ventaja)** es una proporción dividida por su complementario
 - Odd = $p / (1 - p)$ expresa cuánto más probable es que se produzca un hecho frente a que no se produzca. Es una forma alternativa de expresar una probabilidad
- Se define el **Odds Ratio** como el cociente del odd de tener la enfermedad en presencia del factor respecto al odd de tenerla en ausencia del factor
 - Se puede expresar el cociente entre 2 probabilidades (RR), y entre 2 odds (OR)

$$OR = \frac{\frac{P_{F=1}(E=1)}{1-P_{F=1}(E=1)}}{\frac{P_{F=0}(E=1)}{1-P_{F=0}(E=1)}} = \frac{\frac{a}{b}}{\frac{c}{d}} = \frac{a \cdot d}{b \cdot c}$$

	E=1	E=0	Total
F=1	a	b	a + b
F=0	c	d	c + d
Total	a + c	b + d	

- Como $0 \leq p \leq 1$, entonces $0 \leq \text{Odd} \leq \text{inf}$ y $0 \leq OR \leq \text{inf}$
- Si la enfermedad es rara, entonces **RR ≈ OR**
 - Como a y c son pequeños, en la fórmula de RR aproximamos $(a+b) \approx b$ $(c+d) \approx d$

Ejemplo. Riesgo Relativo y Odds Ratio

- Ejemplo

		Bajo Peso		Total	Total
Fumador	$E=0$	$E=1$			
		$F=0$	$F=1$	a	b
		86	29	115	$a + b$
	$F=1$	44	30	74	$c + d$
Total		130	59		$a + c$
				Total	$b + d$

$$OR = \frac{P_{F=1}(E=1) / P_{F=0}(E=1)}{1 - P_{F=1}(E=1) / 1 - P_{F=0}(E=1)} = \frac{\frac{30}{74} / \frac{44}{74}}{\frac{29}{115} / \frac{86}{115}} = \frac{0.405 / 0.595}{0.252 / 0.748} = \frac{0.682}{0.337} = 2.022$$

$$RR = \frac{P_{F=1}(E=1)}{P_{F=0}(E=1)} = \frac{30 / 74}{29 / 115} = \frac{0.405}{0.252} = 1.608$$

Ejemplo: Riesgo relativo (RR) y odds ratio (OR)

```
> t1 <- table ( xx$fumador, xx$bajo_pes)
> t1
  0  1
0 86 29
1 44 30
> a <- t1[2,2]    ## E=1  F=1
> b <- t1[2,1]    ## E=0  F=1
> c <- t1[1,2]    ## E=1  F=0
> d <- t1[1,1]    ## E=0  F=0
> ## Riesgo Relativo (RR)
> RR <- (a/(a+b)) / (c/(c+d))
> RR
[1] 1.607642
> ## Odds Ratio (OR)
> OR <- (a*d) / (b*c)
> OR
[1] 2.021944
```

	E=1	E=0	Total
F=1	a	b	a + b
F=0	c	d	c + d
Total	a + c	b + d	

- **Riesgo Relativo:** la proporción de mujeres con niños con bajo peso al nacer es **1.6 veces superior** en el grupo de fumadoras que en el grupo de no fumadoras
- **Odds Ratio:** la proporción de mujeres con niños con bajo peso al nacer con respecto a las que tuvieron niños con peso normal es **2 veces superior** en el grupo de fumadoras que en el grupo de no fumadoras
- En R hay una librería especializada en cálculo de medidas epidemiológicas: **epitools**

Ejercicio

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
- Describe **la tabla de contingencia** entre RACE y DFREE
 - Frecuencias absolutas y porcentajes: cómo se distribuye DFREE en las categorías de RACE
 - Gráficos de sectores y barras
- Estudiar con el **test de chi-cuadrado** si hay asociación entre DFREE y las variables independientes categóricas: IVHX, RACE, TREAT, SITE
- Crear un **informe** con las 4 tablas de contingencia ejecutando la función AnalisisBinariaCategorica

Estadística Aplicada a la Investigación Biomédica con R

8 Inferencia básica de Variables Continuas

- ✓ Comparación de variables continuas. Esquema general
- ✓ Tests para dos muestras independientes
- ✓ Tests para una muestra
- ✓ Tests para datos apareados
- ✓ Análisis de muchas variables

Comparación variable continua en varias muestras

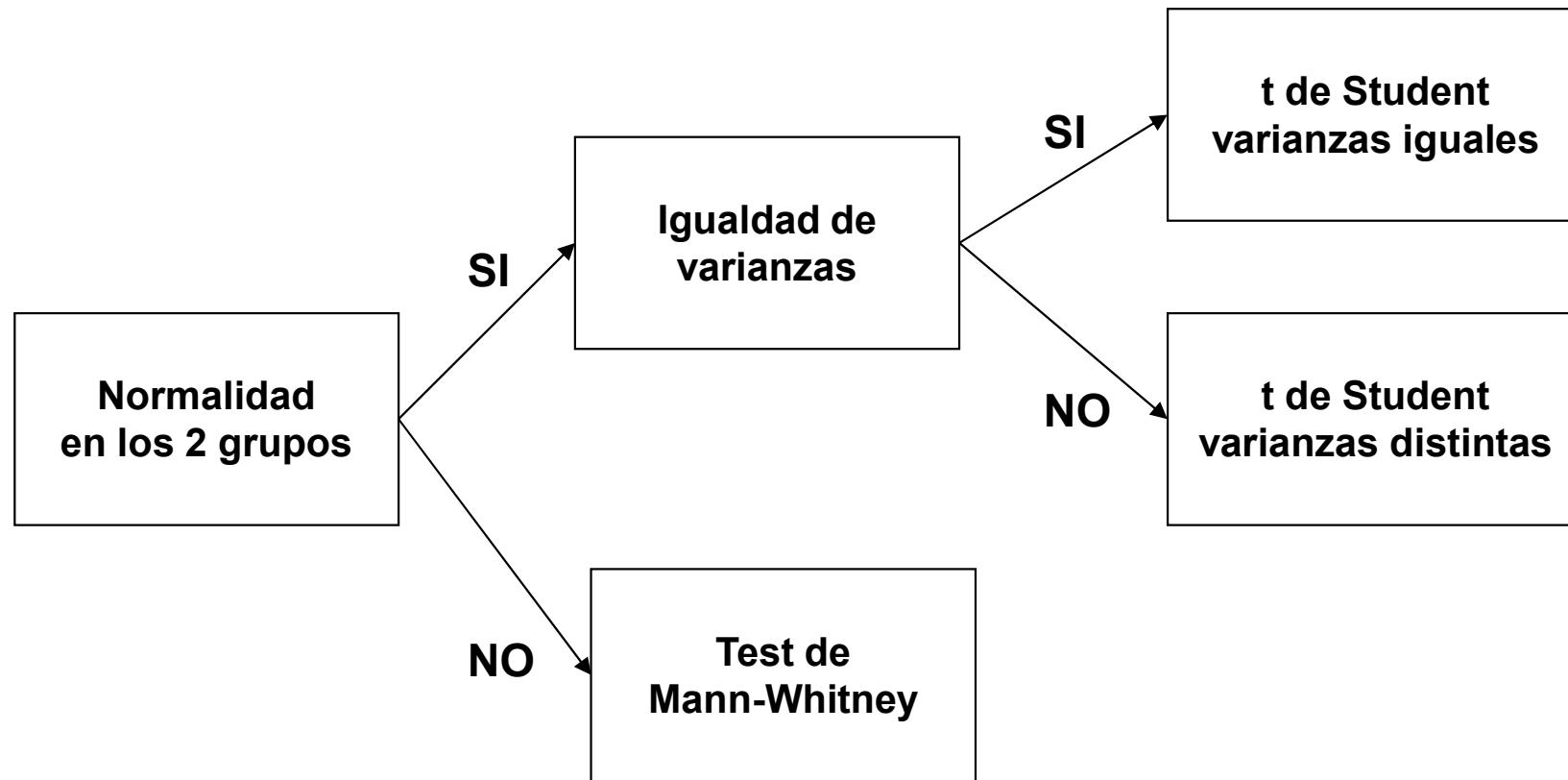
- Se desea contrastar si los valores de **una variable continua** son iguales en **varios grupos**, definidos por las categorías de una variable categórica

	K=2 grupos	K>2 grupos
Normalidad	t de Student	ANOVA
No normalidad	Mann-Whitney Wilcoxon	Kruskal-Wallis

- Las **pruebas para distribuciones normales** (t de Student / ANOVA) contrastan **la media** de la variable en 2 o más grupos
- Las **pruebas no paramétricas** son más generales contrastando si las **distribuciones** son iguales

Comparación de una variable continua en 2 muestras

- El esquema de análisis cuando deseamos contrastar los valores de una variable continua en **2 grupos** es el siguiente:



Test t de Student para 2 muestras independientes

- El **test de t de Student** contrasta si la **media** de una variable continua es igual en dos **muestras independientes**
 - $H_0: \mu_1 = \mu_2$
 - $H_1: \mu_1 \neq \mu_2$
 - Cuando **las varianzas son iguales** en ambas muestras, el contraste está basado en el estadístico:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

que bajo la hipótesis nula sigue una t de Student con n_1+n_2-2 gl

$$s = \sqrt{\frac{(n_1-1) \cdot s_1^2 + (n_2-1) \cdot s_2^2}{(n_1+n_2-2)}}$$

s es un estimador de la SD de toda la muestra

- El **test de t de Student** para muestras **con varianzas distintas**, se basa en

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

s_1 y s_2 son estimadores de las SD en cada grupo

Test de contraste de igualdad de varianzas

- Previamente al test de t de Student, se realizará un **contraste de igualdad de varianzas**
 - $H_0: \sigma_1 = \sigma_2$
 - $H_1: \sigma_1 \neq \sigma_2$
- Este contraste está basado en el cociente σ_1 / σ_2 que será igual o parecido a 1 si ambas varianzas son iguales.
- El **test de Levene** para igualdad de varianzas está en el paquete *car* de R
 - Es menos sensible cuando no hay normalidad

Test de Mann-Whitney 2 muestras independientes

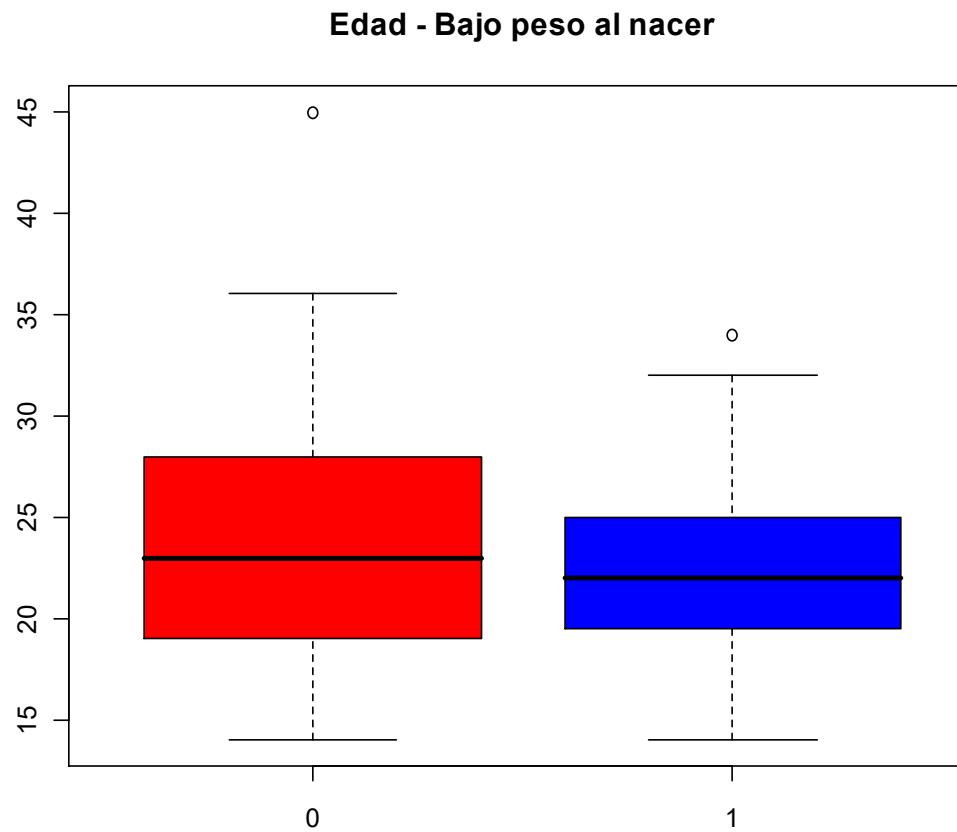
- El **test de Mann-Whitney** o **test de Wilcoxon** contrasta **si las distribuciones** de 2 muestras independientes **son iguales**
 - H_0 : distribuciones iguales
 - H_1 : distribuciones distintas
- Se dice que es un **test no paramétrico**, porque no hace supuestos sobre la distribución de la variable continua
- Es un test basado en **rangos**
 - Se ordenan las 2 muestras juntas, y se asignan rangos, el orden en que aparece cada observación
 - El estadístico de contraste se basa en **la suma de rangos** de una de las 2 muestras, y es una aproximación para muestras grandes

$$Z = \frac{S_1 - n_1(n_1 + n_2 + 1)/2}{\sqrt{n_1 n_2 (n_1 + n_2 + 1)/12}}$$

que sigue una distribución normal estándar

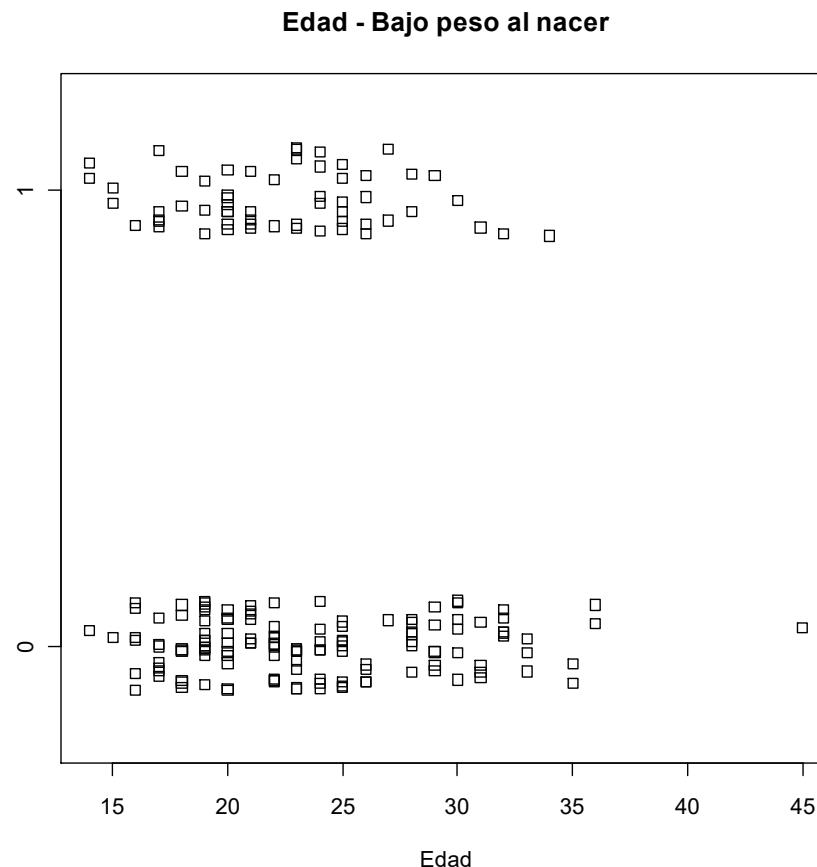
Ejemplo: Comparación 2 muestras. Boxplots

```
> ## Boxplot  
> dev.new()  
> boxplot ( xx$edad ~ xx$bajo_pes, col=c("red","blue") ,  
+           main="Edad - Bajo peso al nacer" )
```



Ejemplo: Comparación 2 muestras. Stripchart

```
> ## Stripchart (la opción "jitter" es porque no se sabe cuántas observaciones hay  
> ## en cada punto). Tiene sentido para muestras pequeñas  
> dev.new()  
> stripchart( xx$edad ~ xx$bajo_pes, method="jitter" , xlab="Edad",  
+               main="Edad - Bajo peso al nacer" )
```



Ejemplo: Comparación 2 muestras. Normalidad

```
> ## Contrastes de Normalidad en cada grupo
> lillie.test ( xx$edad[xx$bajo_pes==0] )
    Lilliefors (Kolmogorov-Smirnov) normality test
data: xx$edad[xx$bajo_pes == 0]
D = 0.1093, p-value = 0.000634

> lillie.test( xx$edad[xx$bajo_pes==1] )
    Lilliefors (Kolmogorov-Smirnov) normality test
data: xx$edad[xx$bajo_pes == 1]
D = 0.0884, p-value = 0.301

> shapiro.test ( xx$edad[xx$bajo_pes==0])
    Shapiro-Wilk normality test
data: xx$edad[xx$bajo_pes == 0]
W = 0.9497, p-value = 0.0001080

> shapiro.test ( xx$edad[xx$bajo_pes==1])
    Shapiro-Wilk normality test
data: xx$edad[xx$bajo_pes == 1]
W = 0.9818, p-value = 0.521
```

- Se acepta la hipótesis de normalidad en solo uno de los grupos
- Por tanto, es preferible el test no paramétrico de Mann-Whitney al de t de Student

Ejemplo: Igualdad de varianzas

```
> ## Test de igualdad de varianza
> var.test ( xx$edad ~ xx$bajo_pes)

    F test to compare two variances

data: xx$edad by xx$bajo_pes
F = 1.5323, num df = 129, denom df = 58, p-value = 0.06885
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.966926 2.338961
sample estimates:
ratio of variances
      1.532254

>
> library(car)
> leveneTest( xx$edad ~ xx$bajo_pes)
Levene's Test for Homogeneity of Variance (center = median)
  Df F value Pr(>F)
group   1  2.7021 0.1019
     187
```

- Se acepta que la variable edad tiene la misma varianza en los dos grupos

Ejemplo: Test de t de Student. Igualdad de medias

```
> ## Test de igualdad de medias. T de Student
> t.test ( xx$edad ~ xx$bajo_pes)

    Welch Two Sample t-test

data: xx$edad by xx$bajo_pes
t = 1.7737, df = 136.941, p-value = 0.07834
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.1558349 2.8687423
sample estimates:
mean in group 0 mean in group 1
23.66154      22.30508

> t.test ( xx$edad ~ xx$bajo_pes, var.equal=T)

    Two Sample t-test

data: xx$edad by xx$bajo_pes
t = 1.6381, df = 187, p-value = 0.1031
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.2770981 2.9900056
sample estimates:
mean in group 0 mean in group 1
23.66154      22.30508
```

Ejemplo: Test de Mann-Whitney

```
> ## Test no paramétrico de Mann-Whitney o test de Wilcoxon
> wilcox.test ( xx$edad ~ xx$bajo_pes)

  Wilcoxon rank sum test with continuity correction

data: xx$edad by xx$bajo_pes
W = 4238, p-value = 0.2471
alternative hypothesis: true location shift is not equal to 0
```

- Tanto el test de t de Student ($P=0.103$) y el de Mann-Whitney ($P=0.247$) nos indican que no hay diferencia significativas en la edad de la madre en los 2 grupos (las que tuvieron niños con peso normal y las que lo tuvieron con bajo peso)

Tests para una muestra

- El **test de t de Student** para una muestra contrasta si **la media** de una variable continua que sigue una **distribución normal** es igual a un determinado valor μ_0
 - $H_0: \mu = \mu_0$
 - $H_1: \mu \neq \mu_0$
 - el contraste está basado en **el estadístico**:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

\bar{x} y s son la media y la desviación estándar de la muestra

- Al igual que para 2 muestras, si los datos no se distribuyen según una distribución normal, se aplica un **test no paramétrico de Mann-Whitney** basado en rangos

Ejemplo: Tests para una muestra

```
> ## Contraste de Normalidad
> lillie.test ( xx$edad )

D = 0.0945, p-value = 0.000303
> shapiro.test ( xx$edad )
W = 0.9598, p-value = 3.19e-05
> ## T de Student
> t.test ( xx$edad , mu=24 )
    One Sample t-test

data: xx$edad
t = -1.9768, df = 188, p-value = 0.04953
alternative hypothesis: true mean is not equal to 24
95 percent confidence interval:
 22.47779 23.99840
sample estimates:
mean of x
 23.23810

> ## Test no paramétrico de Mann-Whitney o test de Wilcoxon
> wilcox.test ( xx$edad, mu=24 )
    Wilcoxon signed rank test with continuity correction

data: xx$edad
V = 6147.5, p-value = 0.01522
alternative hypothesis: true location is not equal to 24
```

Tests para datos apareados

- El **test de t de Student para dos muestras apareadas o relacionadas** contrasta si **la media** de una variable continua es igual en dos muestras que no son independientes
 - $H_0: \mu_1 = \mu_2$
 - $H_1: \mu_1 \neq \mu_2$
 - Se aplica cuando cada observación de una muestra está relacionada con otra de la otra muestra. Por ejemplo, una variable que se ha medido en los mismos individuos en dos momentos del tiempo
 - El contraste está basado en **el estadístico**:

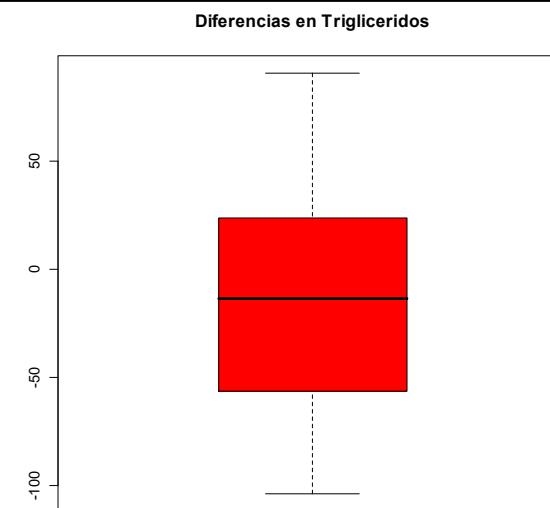
$$t = \frac{\bar{d}_1}{s_d / \sqrt{n}}$$

\bar{d}_1 y s_d son la media y la desviación estándar de **las diferencias**

- Si los datos no se distribuyen según una distribución normal, se aplica un **test no paramétrico de Mann-Whitney** para muestras apareadas, basado en rangos, que contrasta si las distribuciones son iguales

Ejemplo: Tests para datos apareados

```
> ## Datos de Trigliceridos de 16 pacientes, antes y después de una dieta
> trigl.basal <- c(159,93,130,174,148,148,85,180,92,89,204,182,110,88,134,84)
> trigl.final <- c(194,122,158,154,93,90,101,99,183,82,100,104,72,108,110,81)
> trigl.dif <- trigl.final - trigl.basal
> trigl.dif
[1]   35    29    28   -20   -55   -58    16   -81    91    -7 -104   -78   -38    20   -24   -3
>
> ## Boxplot
> dev.new()
> boxplot ( trigl.dif, col=c("red","blue"), main="Diferencias en Trigliceridos" )
>
> ## Contrastes de Normalidad de las diferencias
> lillie.test( trigl.dif )
D = 0.1039, p-value = 0.916
> shapiro.test ( trigl.dif)
W = 0.9761, p-value = 0.925
```



Ejemplo: Tests para datos apareados

```
> ## Test de igualdad de medias para muestras pareadas. T de Student
> t.test ( trigl.basal, trigl.final, paired=T )

  Paired t-test

data:  trigl.basal and trigl.final
t = 1.2019, df = 15, p-value = 0.2480
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-12.0368 43.1618
sample estimates:
mean of the differences
15.5625

> ## Test no paramétrico de Mann-Whitney o test de Wilcoxon para muestras pareadas
> wilcox.test ( trigl.basal, trigl.final, paired=T )

  Wilcoxon signed rank test with continuity correction

data:  trigl.basal and trigl.final
V = 89.5, p-value = 0.2774
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(trigl.basal, trigl.final, paired = T) :
  cannot compute exact p-value with ties
```

- Se acepta la hipótesis nula de igualdad de la variable en los 2 momentos ($P=0.248$)

Ejemplo: Análisis de muchas variables

```
> ## ALLSubset contiene 1000 variables genéticas y una que define 2 clases (última col)
> xx <- read.delim( "C://Bioestadistica con R/Datos/ALLSubset.txt", sep=" ")
> dim(xx)
[1] 79 1001
> head(names(xx))
[1] "X1009_at" "X1010_at" "X1012_at" "X1069_at" "X1081_at" "X1084_at"
> tail(names(xx))
[1] "AFFX.LysX.3_at"    "AFFX.M27830_5_at"  "AFFX.MurFAS_at"    "AFFX.MurIL4_at"
[5] "AFFX.ThrX.M_at"    "mol.biol"
> table( xx$mol.biol )
BCR/ABL      NEG
      37       42
> ## Análisis de la 1ª variable
> tt.out = t.test ( xx[, 1] ~ xx$mol.biol )
> tt.out
    Welch Two Sample t-test
data: xx[, 1] by xx$mol.biol
t = 2.3769, df = 75.371, p-value = 0.02
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.03713917 0.42152137
sample estimates:
mean in group BCR/ABL      mean in group NEG
         9.343180            9.113849
```

- Se pretende almacenar las **medias** de **1000 variables genéticas** para los 2 grupos definidos por la variable “*mol.biol*” y el resultado del test de **t de Student**

Ejemplo: Análisis de muchas variables

```
> names(tt.out)
[1] "statistic"    "parameter"     "p.value"        "conf.int"       "estimate"       "null.value"
[7] "alternative"   "method"        "data.name"
>
> tt.out$p.value
[1] 0.01999904
> tt.out$estimate
mean in group BCR/ABL      mean in group NEG
         9.343180              9.113849
>
> ## Análisis de TODAS las variables
> p.values.tt = rep ( NA, ncol(xx)-1 )
> means.gr     = matrix ( NA, ncol(xx)-1 , 2 )
>
> for ( i in 1:(ncol(xx)-1) )
+ {
+   tt.out = t.test ( xx[ , i ] ~ xx$mol.biol )
+   p.values.tt [ i ] = tt.out$p.value
+   means.gr     [ i , ] = tt.out$estimate
+ }
```

- Los objetos `$p.value` y `$estimate` contienen los resultados para almacenar
- Se define un vector para almacenar los P-Values que tiene el número de variables como longitud; y una matriz para las medias, con ese mismo número de filas y 2 columnas
- En cada iteración del bucle definido por la orden `for` se realiza el análisis de una variable

Ejemplo: Análisis de muchas variables

```
> head(p.values.tt)
[1] 0.01999904 0.13736624 0.86576064 0.03257377 0.24616155 0.37476374
> head(means.gr)
      [,1]     [,2]
[1,] 9.343180 9.113849
[2,] 5.225216 5.320958
[3,] 3.696990 3.708563
[4,] 3.920738 3.703818
[5,] 8.289516 8.515055
[6,] 4.506648 4.564573
> ## Exploramos los resultados
> min ( p.values.tt )
[1] 1.605211e-06
> col.min = which.min ( p.values.tt )
> names(xx) [ col.min ]
[1] "X37403_at"
> length ( p.values.tt [ p.values.tt < 0.001 ] )
[1] 46
> col.001 = which ( p.values.tt < 0.001 )
> col.001
[1]    9   18   31  123  177  194  261  265  268  273  275  319  377  379  409  420  458  459
[19] 468  489  490  571  578  602  615  646  659  660  673  710  725  728  744  745  767  776
[37] 833  846  860  873  882  898  917  920  921  967
```

- Se pueden **explorar los resultados** del análisis, y seleccionar los más interesantes
- Por ejemplo, las variables con p-valores < 0.001 o la más significativa

Ejercicio

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
- Realizar 3 **boxplots** de las variables independientes continuas (AGE, BECK, NDRUGTX) en los dos grupos definidos por la variable DFREE (0 y 1)
- Estudiar para la variable BECK si hay diferencias significativas en los 2 grupos de la variable DFREE:
 - Contrastar la **normalidad** con el test de Kolmogorov-Smirnov ($N>50$) para decidir qué test es adecuado
 - Test de **igualdad de varianzas**
 - **Tests de t de Student y de Wilcoxon**

Estadística Aplicada a la Investigación Biomédica con R

9 Análisis de la Varianza

- ✓ **Análisis de la varianza de un factor**
- ✓ **Comparaciones por pares**
- ✓ **Modelos factoriales**

Comparación variable continua en varias muestras

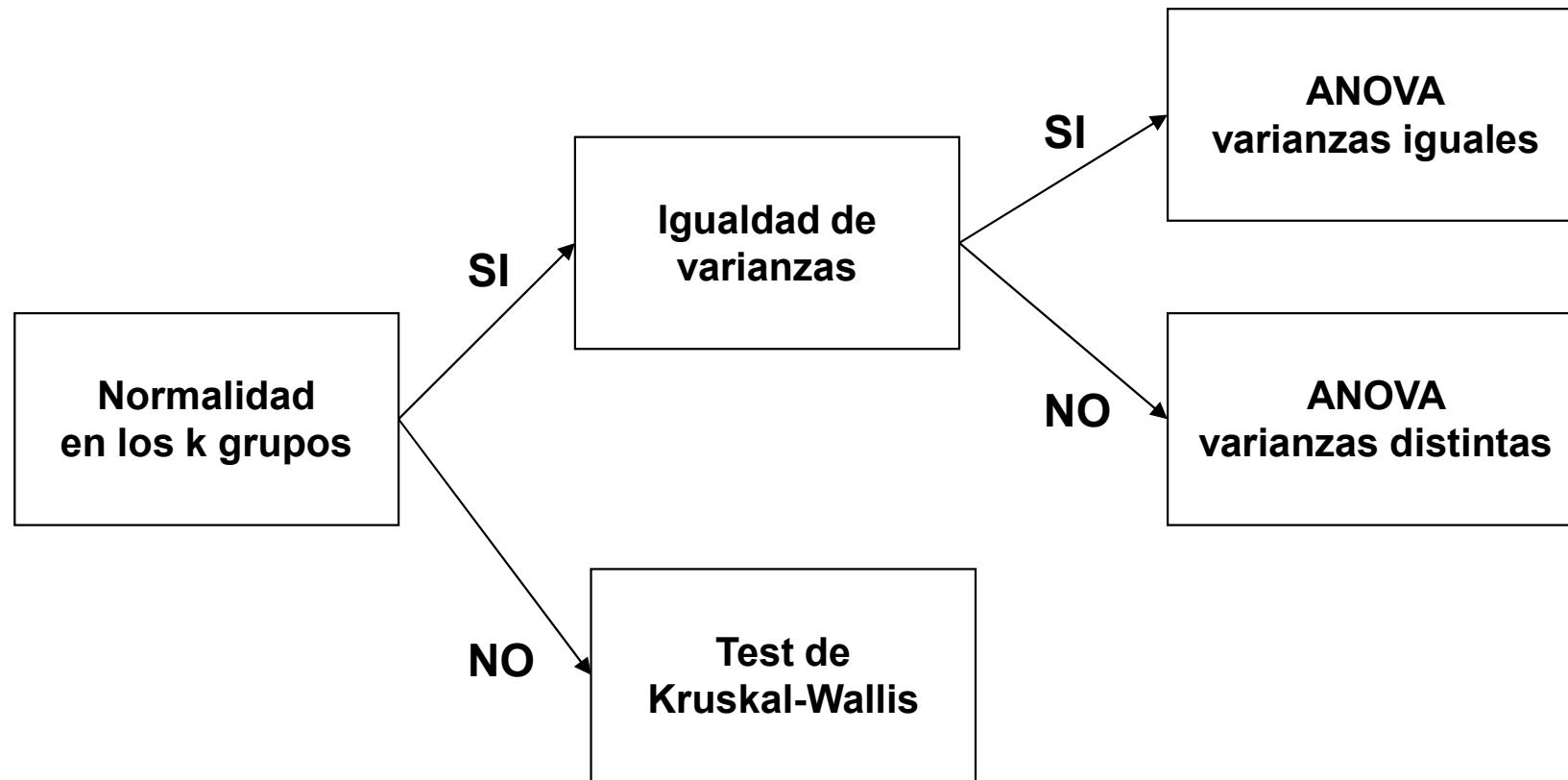
- Se desea contrastar si los valores de **una variable continua** son iguales en varios grupos, definidos por las categorías de una variable categórica

	K=2 grupos	K>2 grupos
Normalidad	t de Student	ANOVA
No normalidad	Mann-Whitney Wilcoxon	Kruskal-Wallis

- Las **pruebas para distribuciones normales** (t de Student / ANOVA) contrastan **la media** de la variable en 2 o más grupos
- Las **pruebas no paramétricas** son más generales contrastando si las **distribuciones** son iguales

Comparación de una variable continua en k muestras

- El esquema de análisis cuando deseamos contrastar los valores de una variable continua en **k grupos ($k>2$)** es el siguiente:



Análisis de la varianza de un factor

- El **análisis de la varianza (ANOVA, analysis of variance)** contrasta si **la media** de una variable continua es igual en k **muestras independientes**
 - $H_0: \mu_1 = \mu_2 = \dots = \mu_k$
 - $H_1: \mu_i \neq \mu_j$ para i, j
 - La **hipótesis alternativa** es que al menos las medias de 2 grupos son distintas
 - El **test de t de Student** es el caso particular del test de ANOVA para 2 grupos

Análisis de la varianza de un factor

- **El modelo del análisis de la varianza** se presenta de la siguiente forma:
 - Sea y_{ij} la observación j-ésima del grupo i. El **modelo** supone:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad \begin{matrix} i = 1, \dots, k \\ j = 1, \dots, n_i \end{matrix}$$

donde:

- μ es la **media global**
 - α_i el **efecto del grupo i** $\mu_i = \mu + \alpha_i$
 - ε_{ij} es el **error aleatorio**, que se supone sigue una distribución normal con media 0 y varianza σ^2
-
- La **hipótesis de igualdad de medias** se puede escribir en términos de **igualdad de efectos**
 - $H_0: \alpha_1 = \alpha_2 = \dots = \alpha_k = 0$

Análisis de la varianza de un factor

- **Supuestos del modelo:**
 - La variable continua debe seguir una **distribución normal** en cada uno de los grupos
 - **Homocedasticidad:** igualdad de las varianzas en todos los grupos, aunque hay una versión del ANOVA para grupos con varianzas distintas
- La **varianza total** en las observaciones se divide en dos:
 - variabilidad debida a la diferencia **entre los grupos (between)**
 - variabilidad **dentro de los grupos (whitin)**

$$\sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2 = \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 + \sum_{i=1}^k n_i (\bar{y}_i - \bar{y})^2$$

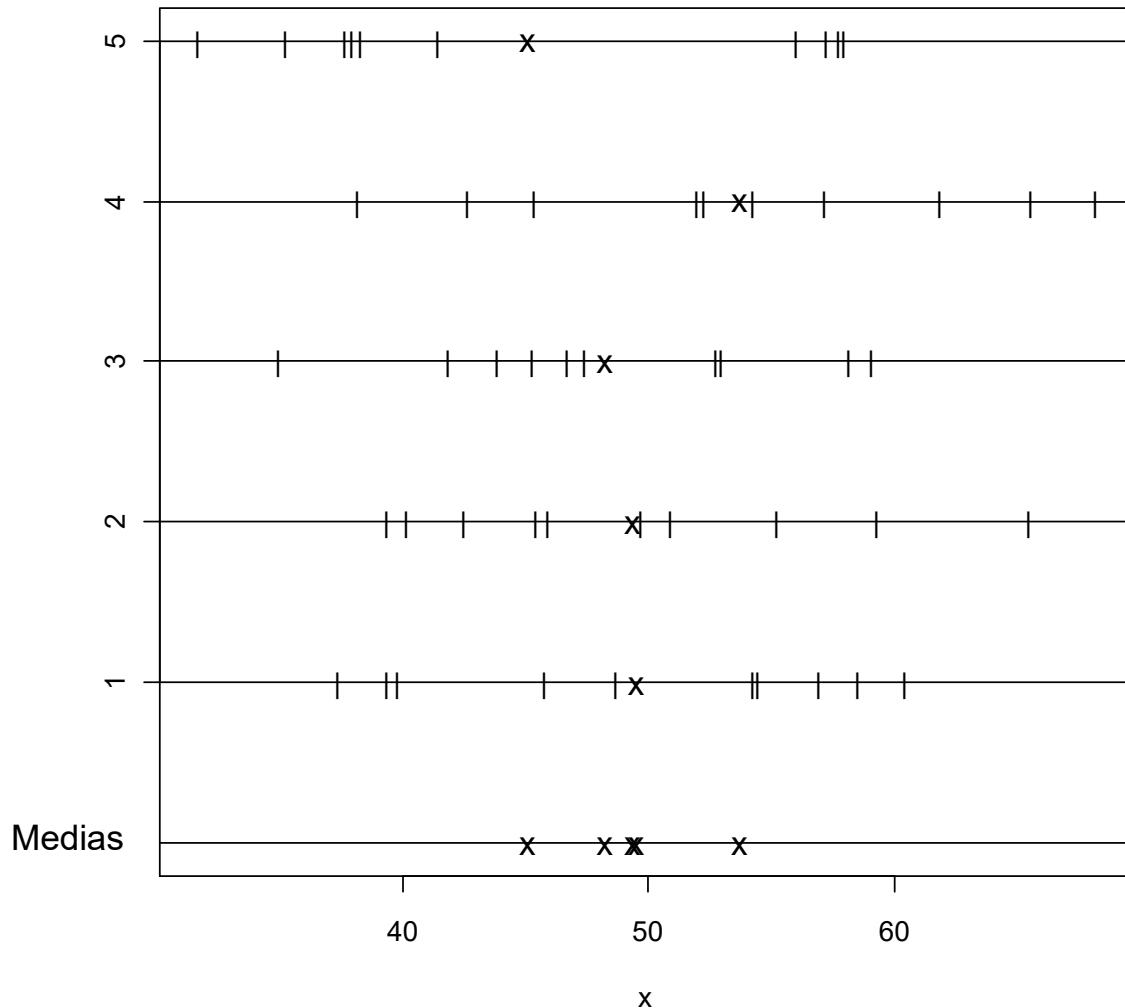
Análisis de la varianza de un factor

- **Tabla del análisis de la varianza**

Fuente Variación	Suma de cuadrados	gl	Medias cuadráticas
Entre grupos	$\sum_{i=1}^k n_i (\bar{y}_i - \bar{y})^2$	k - 1	$MS_B = SS_B / (k-1)$
Dentro de los grupos	$\sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2$	n - k	$MS_w = SS_w / (n-k)$
Total	$\sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2$	n - 1	$MS_T = SS_T / (n-1)$

- Bajo la hipótesis nula de igualdad de medias, **el cociente $F = MS_B/MS_w$** sigue una **distribución F de Snedecor** con $(k-1, n-k)$ grados de libertad
 - Bajo la hipótesis nula, las 2 variabilidades son estimadores de σ^2 y F debería ser 1

Análisis de la varianza de un factor



- Se han generado 5 muestras de 10 obs. de una distribución normal $N(50,10)$
- El test del ANOVA acepta la igualdad de medias con $P=0.623$

```
> summary( aov ( x ~ gr ) )
   Df Sum Sq Mean Sq F value Pr(>F)
gr      1  20.8  20.838  0.2439 0.6236
Residuals 48 4100.3  85.424

> tapply ( x, gr, mean )
 1     2     3     4     5
49.48 49.32 48.22 53.65 45.03
> tapply ( x, gr, sd )
 1     2     3     4     5
8.59  8.54  7.51  9.84 10.73
```

Test de contraste de igualdad de varianzas

- Previamente al análisis de la varianza, se realizará un **contraste de igualdad de varianzas**
 - $H_0: \sigma_1 = \sigma_2 = \dots = \sigma_k$
 - $H_1: \sigma_i \neq \sigma_j$ para i, j
- Test para contrastar la igualdad de varianzas:
 - **Test de Bartlett**
 - Test de Levene
 - Test de Brown–Forsythe
- En el caso de no cumplirse el supuesto de homocedasticidad:
 - Versión del test de **ANOVA con varianzas distintas**
 - Test no paramétrico de Kruskal-Wallis

Test de Kruskal-Wallis k muestras independientes

- El **test de Kruskal-Wallis** contrasta **si las distribuciones** de k muestras independientes **son iguales**
 - H_0 : distribuciones iguales
 - H_1 : distribuciones distintas
- Se dice que es un **test no paramétrico**, porque no hace supuestos sobre la distribución de la variable continua
- Es un test basado en **rangos**
 - Se ordenan las k muestras juntas, y se asignan rangos, el orden en que aparece cada observación
 - El estadístico de contraste se basa en las desviaciones de los rangos en cada grupo con respecto a la media global de los rangos

$$W = (n - 1) \frac{\sum_{i=1}^k n_i (\bar{r}_i - \bar{r})^2}{\sum_{i=1}^k \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2}$$

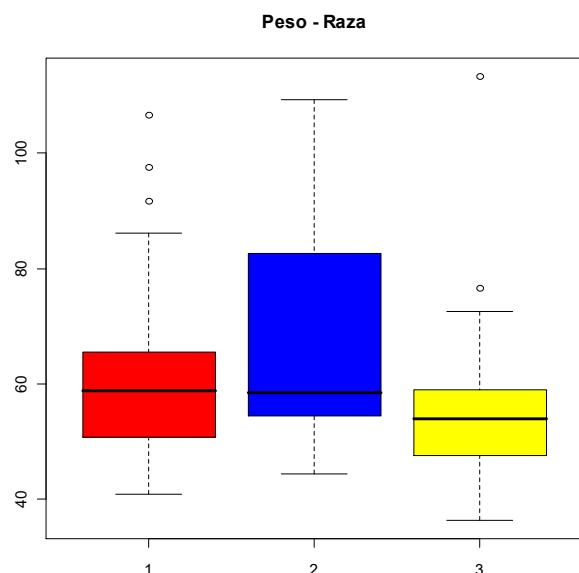
que bajo la hipótesis nula sigue una distribución chi-cuadrado con $k-1$ gl

Comparaciones por pares

- Cuando se rechaza la hipótesis nula de igualdad de medias en el test de ANOVA, la hipótesis alternativa del test es que **al menos 2 de los grupos presentan medias diferentes**
- Se necesita **comparar todos los grupos por pares**
 - **Test t de Student**, si se ha utilizado el test de ANOVA
 - **Test no paramétrico de Wilcoxon**, si se ha utilizado el test de Kruskal-Wallis
- **Corrección de Bonferroni** para comparaciones múltiples
 - Se considera como nivel de significación α / n donde n es el número de comparaciones múltiples, que en este caso es $k(k-1) / 2$
 - A nivel práctico, se considera un **P-valor ajustado**: $P_{\text{ajust}} = 1 - (1 - P_{\text{orig}})^n$

Ejemplo: Comparación k muestras. Descriptivo

```
> ## Boxplots
> dev.new()
> boxplot ( xx$peso ~ xx$raza, col=c("red","blue","yellow"),
+            main="Peso - Raza" )
> ## Medias, medianas y SD
> tapply( xx$peso, xx$raza, mean )
    1      2      3
59.89843 66.59153 54.43841
> tapply( xx$peso, xx$raza, median )
    1      2      3
58.74081 58.51402 53.97805
> tapply( xx$peso, xx$raza, sd )
    1      2      3
13.19687 17.98031 11.39901
```



- Las distribuciones parecen asimétricas, sesgadas a la derecha
- El grupo 3 presenta una media y mediana inferior

Ejemplo: Normalidad e igualdad de varianzas

```
> ## Contrastes de Normalidad en cada grupo (Kolmogorov-Smirnov Test)
> for ( i in 1:3 )
+ {
+   print ( lillie.test( xx$peso[xx$raza==i] ) )
+ }

      Lilliefors (Kolmogorov-Smirnov) normality test
D = 0.1263, p-value = 0.0006781

      Lilliefors (Kolmogorov-Smirnov) normality test
D = 0.2411, p-value = 0.0004408

      Lilliefors (Kolmogorov-Smirnov) normality test
D = 0.142, p-value = 0.001846

> ## Test de igualdad de varianza
> bartlett.test ( xx$peso ~ xx$raza )

      Bartlett test of homogeneity of variances
data: xx$peso by xx$raza
Bartlett's K-squared = 8.3756, df = 2, p-value = 0.01518
```

- La variable peso **no sigue una distribución normal** en los 3 grupos de raza, y no se puede aceptar la hipótesis de varianzas iguales
- Se recomienda usar el **test no paramétrico de Kruskal-Wallis**

Ejemplo: Test de ANOVA de una factor

```
> ## Análisis de la varianza (ANOVA de un factor)
> summary( aov ( xx$peso ~ xx$raza ) )
      Df Sum Sq Mean Sq F value    Pr(>F)
xx$raza        2   2967  1483.74  8.3118 0.0003488 ***
Residuals    186  33203   178.51
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> ## Comparaciones múltiples, dos a dos
> pairwise.t.test( xx$peso , xx$raza , p.adj="bonferroni" )

  Pairwise comparisons using t tests with pooled SD

data: xx$peso and xx$raza

  1     2
2 0.07383 -
3 0.03313 0.00035

P value adjustment method: bonferroni
```

- Si se cumpliesen los supuestos para aplicar el **test de ANOVA** (normalidad y homocedasticidad), obtendríamos que **hay diferencias significativas** de la variable peso en los 3 grupos de raza ($P=0.0003$), y esas diferencias se encuentran entre los grupos 2 y 3 ($P=0.0003$) y entre el 1 y el 3 ($P=0.0331$), es decir, el grupo 3 es diferente a los grupos 1 y 2, que podemos asumir que son iguales

Ejemplo: Test de ANOVA con varianzas distintas

```
> ## Análisis de la varianza con varianzas distintas
> oneway.test ( xx$peso ~ xx$raza )

  One-way analysis of means (not assuming equal variances)

data: xx$peso and xx$raza
F = 7.1525, num df = 2.000, denom df = 63.458, p-value = 0.001580

> ## Comparaciones múltiples, dos a dos
> pairwise.t.test( xx$peso , xx$raza , p.adj="bonferroni" , pool.sd=F )

  Pairwise comparisons using t tests with non-pooled SD

data: xx$peso and xx$raza

  1     2
2 0.257 -
3 0.016 0.009

P value adjustment method: bonferroni
```

- Si se cumpliese el supuesto de normalidad, pero no pudiéramos aceptar la igualdad de varianzas, el **test de ANOVA** nos indicaría que **hay diferencias significativas** de la variable peso en los 3 grupos de raza ($P=0.0016$), y esas diferencias se encuentran entre los grupos 2 y 3 ($P=0.009$) y entre el 1 y el 3 ($P=0.016$)

Ejemplo: Test de Kruskal-Wallis

```
> ## Test no paramétrico de Kruskal-Wallis
> kruskal.test ( xx$peso ~ xx$raza )

  Kruskal-Wallis rank sum test

data: xx$peso by xx$raza
Kruskal-Wallis chi-squared = 13.9605, df = 2, p-value = 0.00093

> ## Comparaciones múltiples, dos a dos (Wilcoxon Test)
> pairwise.wilcox.test( xx$peso , xx$raza , p.adj="bonferroni" )

  Pairwise comparisons using Wilcoxon rank sum test

data: xx$peso and xx$raza

  1     2
2 0.4506 -
3 0.0107 0.0033

P value adjustment method: bonferroni
```

- El **test de Kruskal-Wallis**, que es el más adecuado en esta situación, nos indica que **hay diferencias significativas** de la variable peso en los 3 grupos de raza ($P=0.0009$), y esas diferencias se encuentran entre los grupos 2 y 3 ($P=0.0033$) y entre el 1 y el 3 ($P=0.0107$) (Test de Wilcoxon para contrastar todos los pares)

Modelos factoriales

- **El modelo del análisis de la varianza** se puede extender a más de un factor. El modelo para **2 factores** se presenta de la siguiente forma:
 - Sea y_{ijk} la observación k-ésima del grupo i en el primer factor, y j en el segundo factor. El **modelo** supone:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \quad \begin{matrix} i = 1, \dots, m \\ j = 1, \dots, p \\ k = 1, \dots, n_{ij} \end{matrix}$$

donde:

- μ es la **media global**
- α_i y β_j son los **efectos principales de los 2 factores**
- γ_{ij} es el término de **interacción**
- ε_{ijk} es **el error aleatorio**, que se supone sigue una distribución normal con media 0 y varianza σ^2

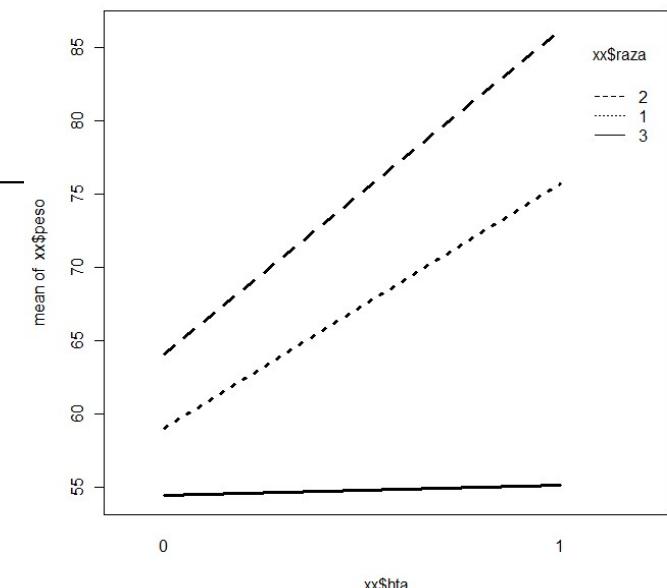
Modelos factoriales

- Se plantean **distintas hipótesis** sobre los efectos principales de ambos factores y sobre la interacción
 - $H_0: \alpha_1 = \alpha_2 = \dots = \alpha_m = 0$
 - $H_0: \beta_1 = \beta_2 = \dots = \beta_p = 0$
 - $H_0: \gamma_{11} = \gamma_{12} = \dots = \gamma_{mp} = 0$
- Se llama **Análisis de la covarianza (ANCOVA)** cuando se incluyen factores categóricos y continuos
 - Son equivalentes a los **Modelos de regresión lineal múltiple**

Ejemplo: ANOVA con 2 factores

```
> ## Modelo de efectos principales
> summary( aov ( xx$peso ~ xx$raza + xx$hta ) )
      Df  Sum Sq Mean Sq F value    Pr(>F)
xx$raza       2   2967.5 1483.74   8.731 0.0002380 ***
xx$hta        1   1764.1 1764.10  10.381 0.0015048 **
Residuals    185 31439.0 169.94
---
> ## Modelos con interacción
> summary( aov ( xx$peso ~ xx$raza * xx$hta ) )
      Df  Sum Sq Mean Sq F value    Pr(>F)
xx$raza       2   2967.5 1483.74   8.8765 0.0002093 ***
xx$hta        1   1764.1 1764.10  10.5538 0.0013796 **
xx$raza:xx$hta 2   849.9  424.97   2.5424 0.0814551 .
Residuals    183 30589.1 167.15
---
> ## Gráfico de la interacción
> dev.new()
> interaction.plot ( xx$hta,xx$raza, xx$peso , lwd=3 )
```

- Hay **diferencias significativas** de la variable peso con raza ($P=0.0002$) y con HTA ($P=0.0014$)
- La **interacción** está “cerca” de la significación ($P=0.0814$), debido a que las mujeres de “otras razas” con HTA presentan un peso inferior al esperado



Ejercicio

- Fichero de datos: umaru.txt
- Estudiar si hay diferencias significativas en la variable BECK (score de depresión) en los 3 grupos definidos por la variable IVHX (historia de consumo de drogas)
 - Medias, medianas, SD de BECK para los 3 grupos
 - **Boxplots** de BECK para los 3 grupos
 - Test de **normalidad**
 - Test de **igualdad de varianzas**
 - **ANOVA** con varianzas iguales (función *aov*)
 - **Test de Kruskal-Wallis**
 - **Comparaciones por pares** de grupos (para el test más adecuado)

Estadística Aplicada a la Investigación Biomédica con R

10 Análisis de Correlaciones

- ✓ Covarianza
- ✓ Correlación lineal de Pearson
- ✓ Correlación no paramétrica

Covarianza

- La **covarianza** es una medida que refleja el **grado de asociación lineal** entre dos variables cuantitativas
- Es una medida de la **variabilidad conjunta** de las dos variables
- La **covarianza** se define como

$$\text{Cov}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

- En el numerador aparece la **suma de productos cruzados (SP_{XY})**
- Cada elemento de la suma de productos cruzados tendrá **signo positivo o negativo** dependiendo de que las observaciones tengan valores mayores o menores que las medias de las variables X e Y
- La covarianza depende de las **unidades de X e Y**

Covarianza

- El **signo** de cada **producto cruzado** depende de que los puntos sean mayores o menores que las medias de las variables X e Y

$(y_i - \bar{y}) > 0$	—	+
\bar{y}		(\bar{x}, \bar{y})
$(y_i - \bar{y}) < 0$	+	—
	$(x_i - \bar{x}) < 0$	$(x_i - \bar{x}) > 0$

Correlación

- La **correlación** es una medida que refleja el **grado de asociación lineal** entre dos variables cuantitativas que **no depende de las unidades** de X e Y
 - Se **estandariza** la covarianza dividiendo por las desviaciones típicas de X e Y
 - A diferencia de la regresión, donde se modela como cambia la variable Y en función de X, en la correlación, **las dos variables** juegan un papel **simétrico**
- Esta medida se conoce como el **coeficiente de correlación de Pearson** y se define como

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

- $-1 < r < 1$

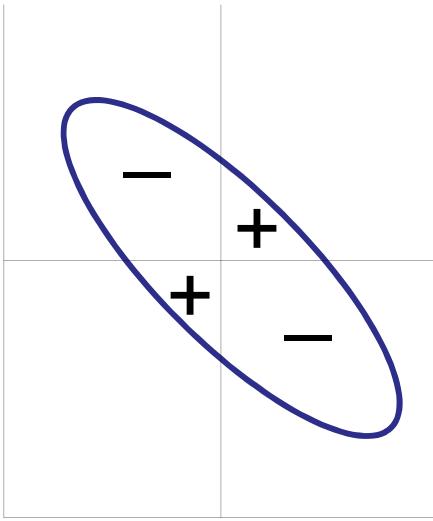
$r = 0$ ausencia de asociación lineal

$r = 1$ asociación perfecta positiva

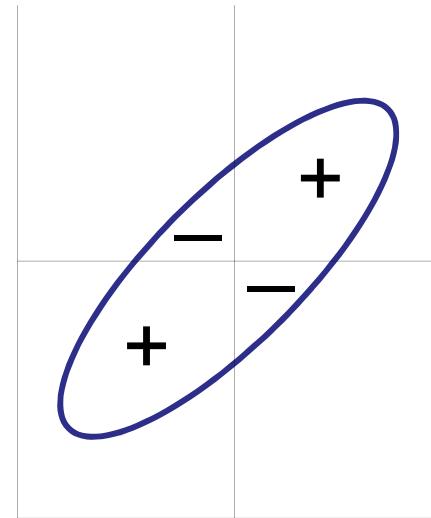
$r = -1$ asociación perfecta negativa

Correlación

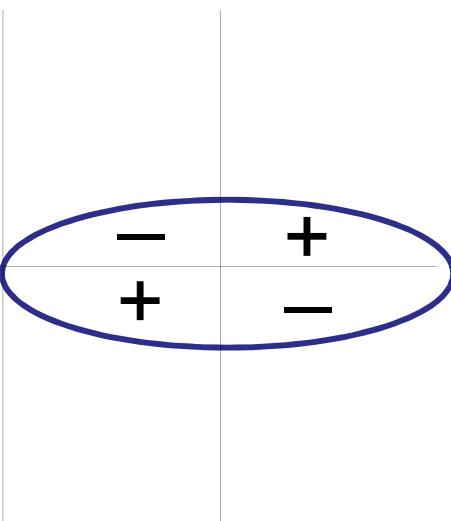
$r < 0$



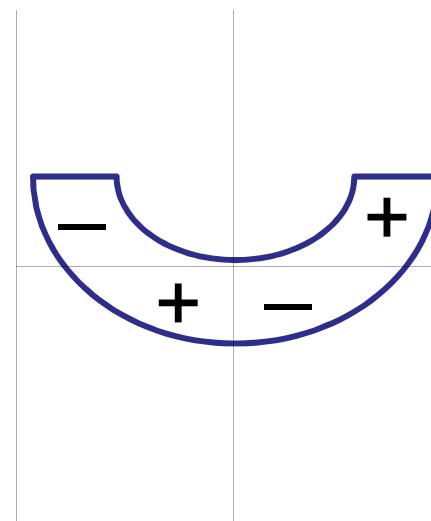
$r > 0$



$r = 0$



$r = 0$



Correlación

- El coeficiente de correlación de Pearson detecta **asociaciones lineales**
 - Si una relación es **monótona** creciente o decreciente, el coeficiente de correlación detecta la asociación, pero la puede **infravalorar** si es **no lineal**
 - No detecta relaciones **cuadráticas**
- El coeficiente de correlación de Pearson **no es una medida robusta**
 - Valores alejados pueden influir mucho en el valor y en el signo de la suma de los productos cruzados
- El coeficiente de correlación de Pearson no se puede utilizar como coeficiente de concordancia
 - **Coeficiente de correlación intraclass (ICC)** para variables cuantitativas
 - **Índice de Kappa** para variables categóricas

Correlación no paramétrica

- El **coeficiente de correlación de Spearman** es una medida de asociación basada en rangos
 - Se sustituyen los valores de las variables por **los rangos**
 - Interpretación similar al coeficiente de Pearson $-1 < \rho < 1$
 - Puede detectar **asociaciones no lineales**, monótonas crecientes o decrecientes

$$\rho = 1 - \frac{\sum_{i=1}^n D_i^2}{n(n^2 - 1)}$$

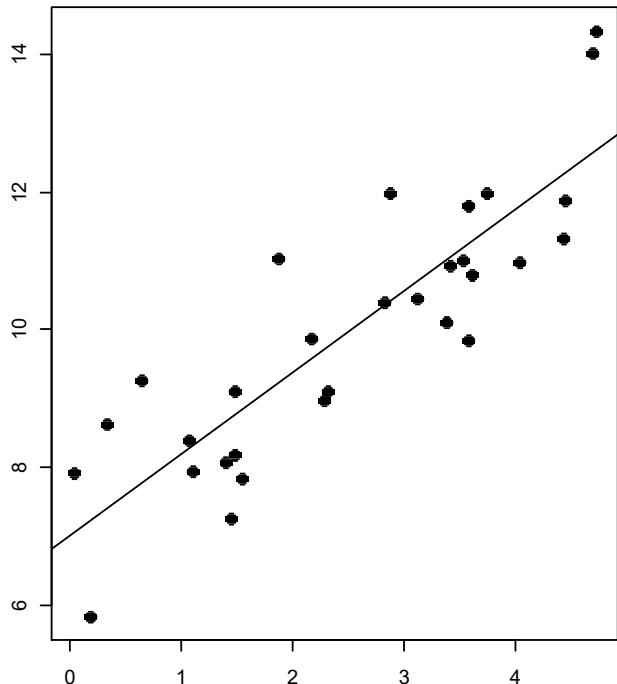
D_i es la diferencia entre los estadísticos de orden de X e Y

- El **coeficiente de correlación de Kendall** está basado en concordancias
 - Se estudia la **concordancia** entre todos los pares (x_i, y_i)

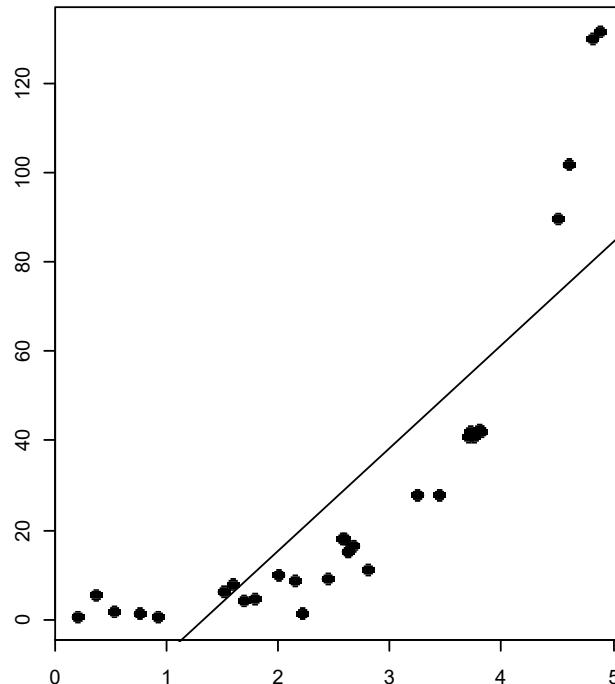
$$\tau = \frac{2 \cdot (N_C - N_D)}{n(n - 1)}$$

N_C y N_D son el número de pares concordantes y discordantes

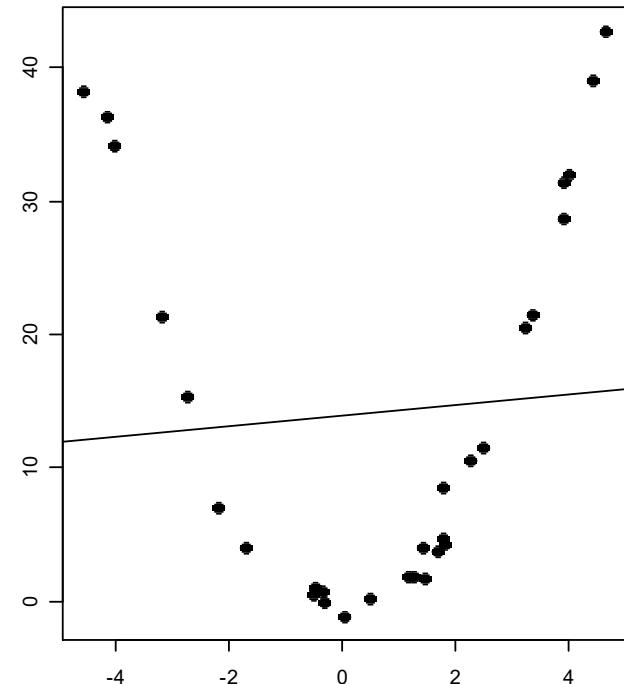
Ejemplo: Coeficientes de correlación



$r = 0.86$
 $\rho = 0.85$



$r = 0.83$
 $\rho = 0.93$



$r = 0.07$
 $\rho = 0.14$

Ejemplo: Correlación

```
> ## Fichero Datos: cystic fibrosis
> xx <- read.csv(file="C://Bioestadistica con R/Datos/cystic fibrosis.csv", sep=";")
> ## Correlación: Pearson, Spearman, Kendall
> cor ( xx$height, xx$weight , method = "pearson", use="pairwise.complete.obs")
[1] 0.9221054
> cor ( xx$height, xx$weight , method = "spearman", use="pairwise.complete.obs")
[1] 0.961894
> cor ( xx$height, xx$weight , method = "kendall", use="pairwise.complete.obs")
[1] 0.8657913
> ## Matriz de correlaciones
> cor ( xx[,4:11], method = "pearson", use="pairwise.complete.obs")
      height     weight      bmp       fev        rv       frc       tlc      pemax
height  1.0000000  0.9221054  0.4407623  0.3166636 -0.5695199 -0.6242769 -0.4595190  0.5992195
weight   0.9221054  1.0000000  0.6703392  0.4492481 -0.6233783 -0.6182327 -0.4213667  0.6362889
bmp      0.4407623  0.6703392  1.0000000  0.5455204 -0.5823729 -0.4343888 -0.3633366  0.2295148
fev      0.3166636  0.4492481  0.5455204  1.0000000 -0.6658557 -0.6651149 -0.4425226  0.4533757
rv       -0.5695199 -0.6233783 -0.5823729 -0.6658557  1.0000000  0.9106029  0.5899035 -0.3155501
frc      -0.6242769 -0.6182327 -0.4343888 -0.6651149  0.9106029  1.0000000  0.7056193 -0.4172078
tlc      -0.4595190 -0.4213667 -0.3633366 -0.4425226  0.5899035  0.7056193  1.0000000 -0.1805401
pemax    0.5992195  0.6362889  0.2295148  0.4533757 -0.3155501 -0.4172078 -0.1805401  1.0000000
```

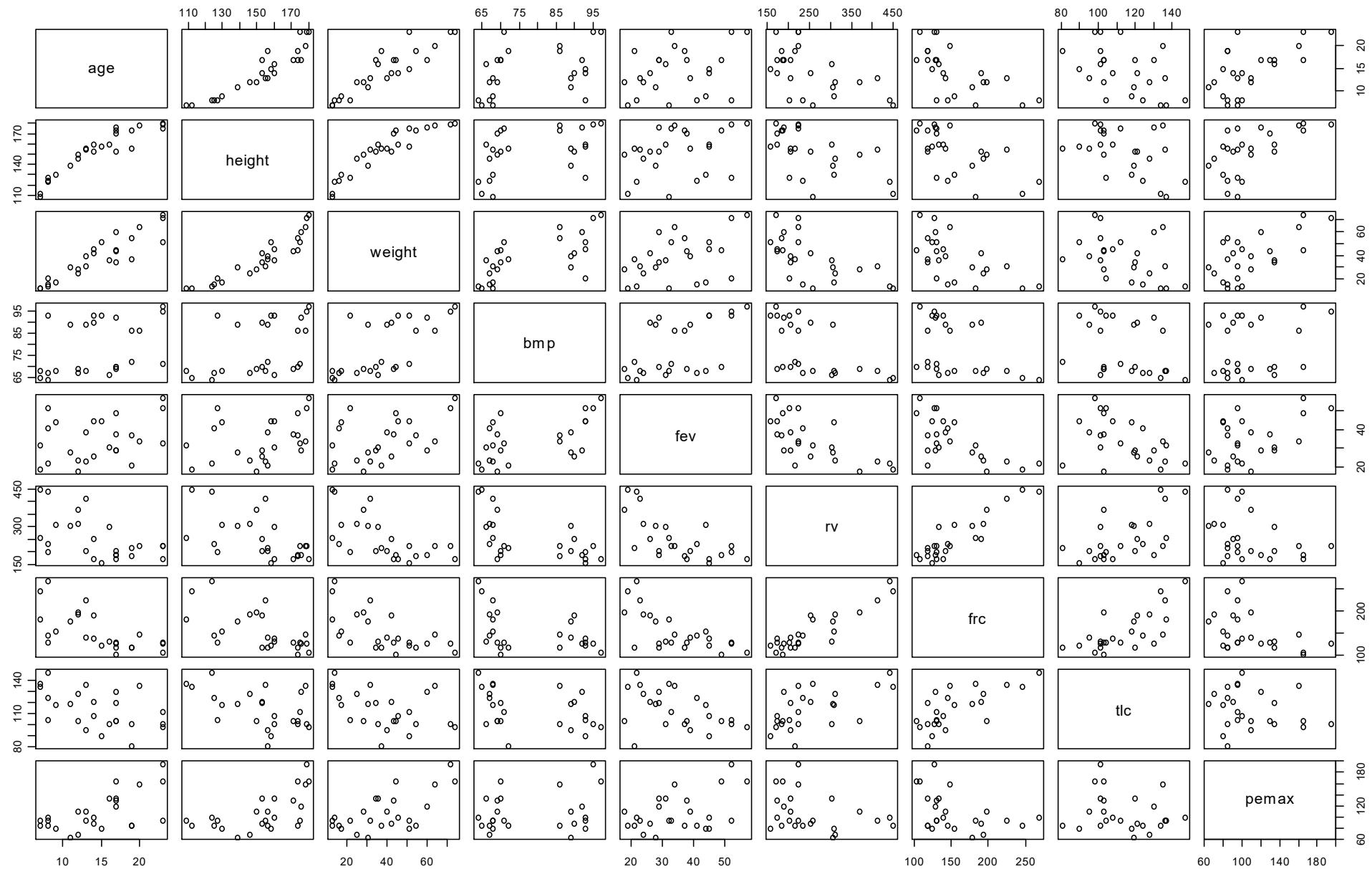
- La función **cor()** calcula los **coeficientes de correlación**. Con el parámetro “*method*=“ podemos elegir el coeficiente de correlación: lineal de Pearson o los no paramétricos de Spearman o Kendall
- La opción *use=“pairwise.complete.obs”* elimina del cálculo los missings en cada pareja de variables. Por defecto, *use=“complete.obs”* que usa casos completos, observaciones que no tienen ningún missing

Ejemplo: Correlación

```
> ## Test del coeficiente de correlación H0: r = 0
> cor.test ( xx$height, xx$weight , method = "pearson", use="pairwise.complete.obs")
  Pearson's product-moment correlation
data: xx$height and xx$weight
t = 11.4288, df = 23, p-value = 5.812e-11
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8290355 0.9654664
sample estimates:
cor
0.9221054
> ## Explorando las asociaciones entre las variables
> dev.new()
> pairs ( xx[ , c(2,4:11) ] )
```

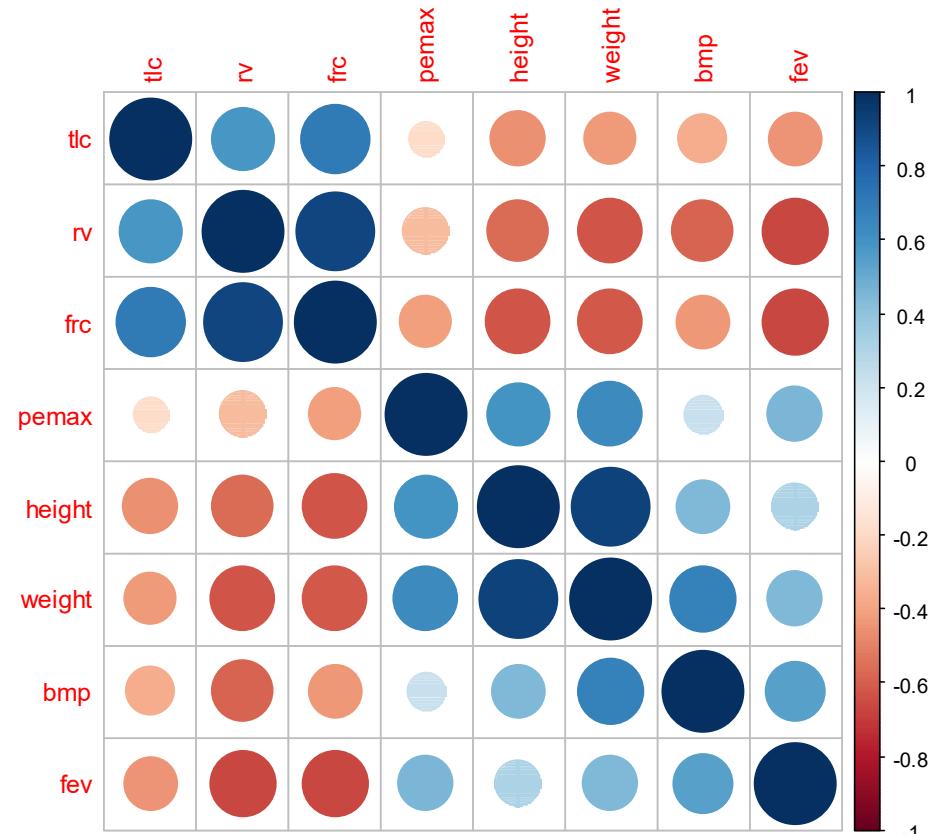
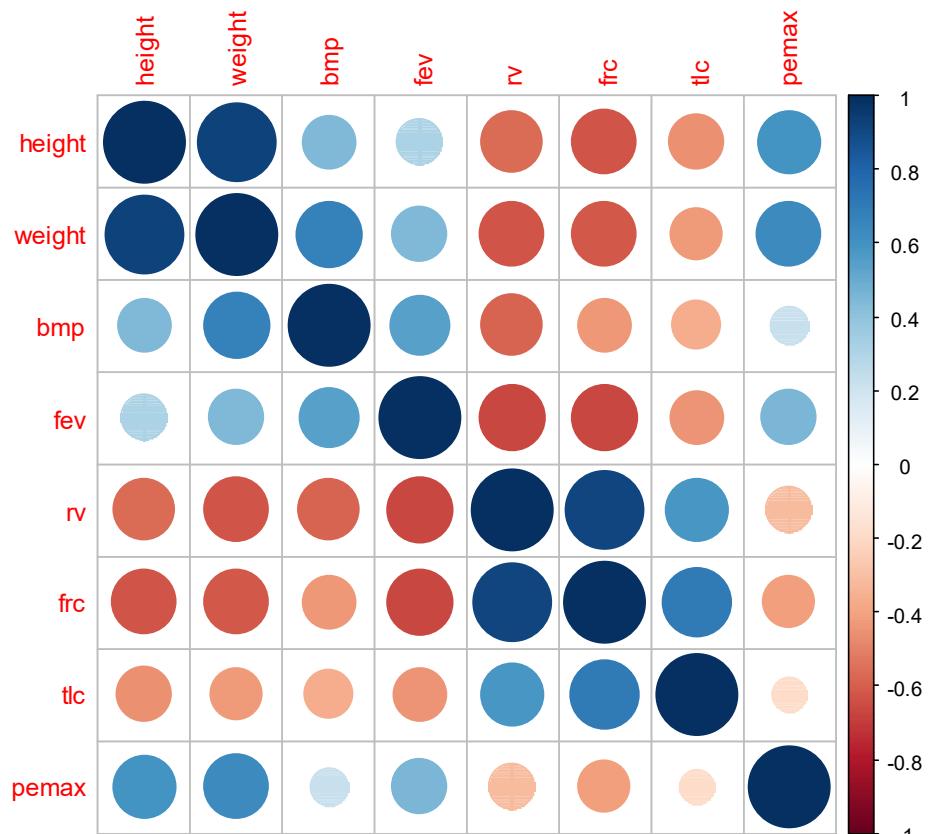
- La función **cor.test()** contrasta si el coeficiente de correlación es igual a 0, lo que indicaría que no hay asociación entre las 2 variables ($H_0: r=0$)
- En este caso, concluimos que la asociación que hay entre las dos variables, altura y peso, es significativa ($P=5.8 \times 10^{-11}$)
- Una función muy útil es **pairs()** que muestra todos los gráficos de dispersión de un conjunto de variables. Es una forma de mostrar gráficamente la matriz de correlaciones

Ejemplo: Gráficos de dispersión



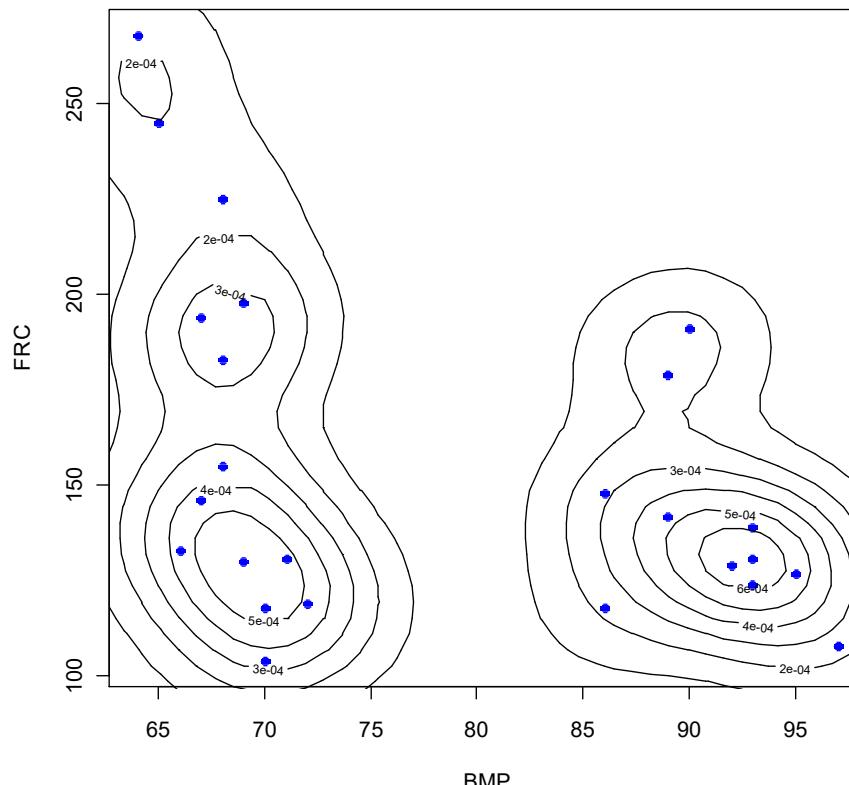
Ejemplo: Matriz de correlaciones

```
> library ( corrplot )
> w.cor = cor ( xx[,4:11], method = "pearson", use="pairwise.complete.obs")
> dev.new()
> corrplot ( w.cor )
> dev.new()
> corrplot ( w.cor , order="hclust" )
```



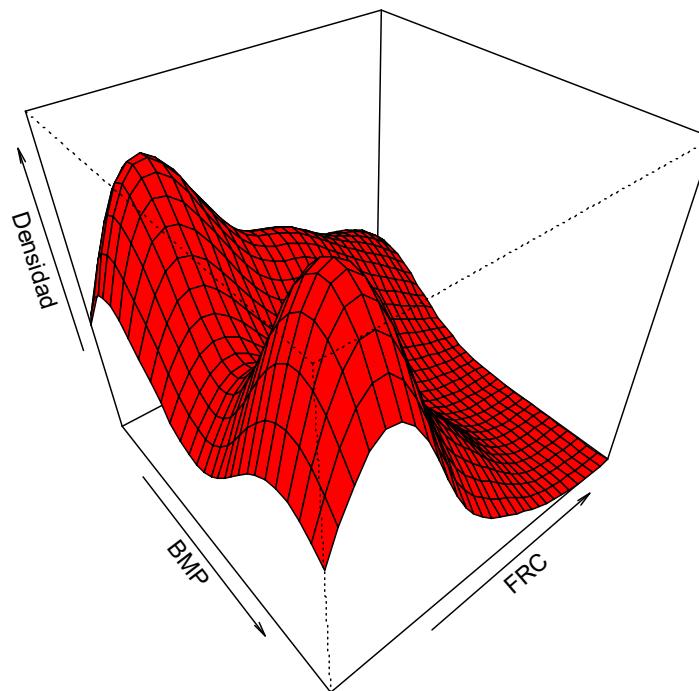
Ejemplo: Gráfico de contorno

```
> library(KernSmooth)
> ## Gráfico de dispersión
> dev.new()
> plot( xx$bmp, xx$frc ,pch=16, xlab="BMP", ylab="FRC", col="blue")
> ## Añadimos las líneas de contorno
> df.graph <- data.frame ( x=xx$bmp, y=xx$frc )
> graph.schema <- bkde2D( df.graph , bandwidth = sapply(df.graph, dpik))
>
> contour (x = graph.schema$x1, y = graph.schema$x2, z = graph.schema$fhat,
+           add=TRUE, lwd=1.2, nlevels=5)
```



Ejemplo: Gráfico de densidad bidimensional

```
> ## Se cargan las funciones
> source ( "C://Bioestadistica con R/R Scripts/Estadística-R Funciones.r")
> ## Se estima la función de densidad bidimensional
> den1 <- bivden( xx$bmp, xx$frc )
>
> ## Gráfico 3D
> dev.new()
> persp (den1$seqx, den1$seqy, den1$den, theta=50, phi=40, col="red",
+         xlab="BMP", ylab="FRC", zlab="Densidad")
```



Ejercicio

- Fichero de datos: Bajo peso al nacer
- Analizar si las variables edad y peso de la madre están asociadas
 - Calcular los **coeficientes de correlación** entre las 2 variables (Pearson, Spearman, Kendall)
 - Contrastar si el coeficiente de correlación de Pearson es igual a 0
 - Realizar un **gráfico de dispersión** de las 2 variables

Estadística Aplicada a la Investigación Biomédica con R

11 Regresión Lineal Simple

- ✓ **Modelos de Regresión**
- ✓ **Regresión lineal simple**
- ✓ **Estimación por mínimos cuadrados**
- ✓ **Tabla ANOVA y pruebas de significación**
- ✓ **Bondad de ajuste**
- ✓ **Regresión lineal robusta**

Modelos de regresión

- Los **modelos de regresión describen la relación** entre una o varias variables predictoras y una variable respuesta mediante una función matemática
 - Y variable dependiente, respuesta, resultado (outcome)
 - X variables independientes, explicativas, predictores, covariables
- **Ajuste de un modelo de regresión**
 - **Describir la relación** entre las variables X e Y mediante una formulación matemática y la **estimación de los parámetros** del modelo
 - Evaluar qué variables predictoras **están relacionadas** con la respuesta Y
 - **Evaluar la magnitud del efecto** de cada variable predictora X en la respuesta Y
 - **Predecir la respuesta Y** a partir de los valores de las variables X

Modelos de regresión

- **Pasos generales** en el ajuste y evaluación de un modelo de regresión

Ajuste del modelo de regresión	<ul style="list-style-type: none">- Estimación de los coeficientes de regresión
Evaluación del modelo	<ul style="list-style-type: none">- Contraste global- Bondad del ajuste- Contrastos de hipótesis de los coeficientes de regresión (P-valores)- Decidir qué variables predictoras están asociadas a la variable respuesta
Evaluación de los efectos de cada variable predictora	<ul style="list-style-type: none">- Cuantificación de los efectos (IC95%)- Interpretación de los resultados

Modelos de regresión

- Los **modelos de regresión** se distinguen por la **variable respuesta**:
 - Si Y es continua modelo de **regresión lineal**
 - Si Y es binaria modelo de **regresión logística binario**
 - Si Y es categórica modelo de **regresión logística multinomial u ordinal**
- Todos los modelos de regresión admiten variables **categóricas o continuas** como **variables independientes**
- Las **estrategias de construcción** de modelos de regresión y el tratamiento de las variables independientes, **son comunes** para todos los modelos
 - Tratamiento de variables **categóricas** con variables dummy
 - Tratamiento de variables **continuas**: categorización, transformaciones, ...
 - **Variables de confusión**
 - **Interacciones**
 - Construcción de **modelos multivariantes**

Regresión lineal simple

- Modelo que representa la **relación lineal** de una variable respuesta Y respecto de una variable predictora X
- El **modelo de regresión lineal simple** se formula como:

$$Y = \beta_0 + \beta_1 \cdot X + \varepsilon$$

- ε es **el error aleatorio**, que se supone sigue una distribución normal con media 0 y varianza desconocida σ^2 que no depende del valor de x (homocedasticidad)
- β_0 y β_1 son los parámetros del modelo, llamados **coeficientes de regresión**: β_0 es el término constante (intercept) y β_1 es la pendiente

Regresión lineal simple

- El modelo de regresión es **lineal respecto a los parámetros**
- Se pueden analizar **relaciones no lineales** con el modelo de regresión lineal, haciendo **transformaciones** en la variable predictora
- Ejemplos:

$$Y = \beta_0 + \beta_1 \cdot \ln(X)$$

$$Y = \beta_0 + \beta_1 \cdot e^X$$

$$Y = \beta_0 e^{\beta_1 X} \Rightarrow \ln(Y) = \ln(\beta_0) + \beta_1 \cdot X$$

Regresión lineal simple

- El modelo de regresión lineal simple también se puede expresar con la **media de Y como una función lineal de X**, que representa una **recta**

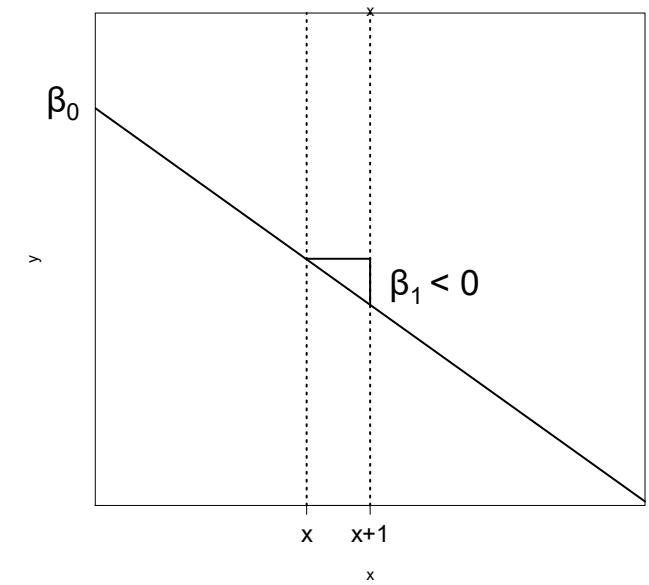
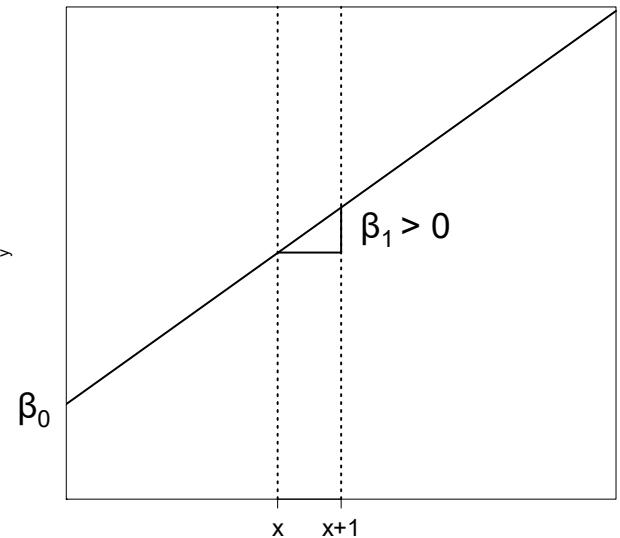
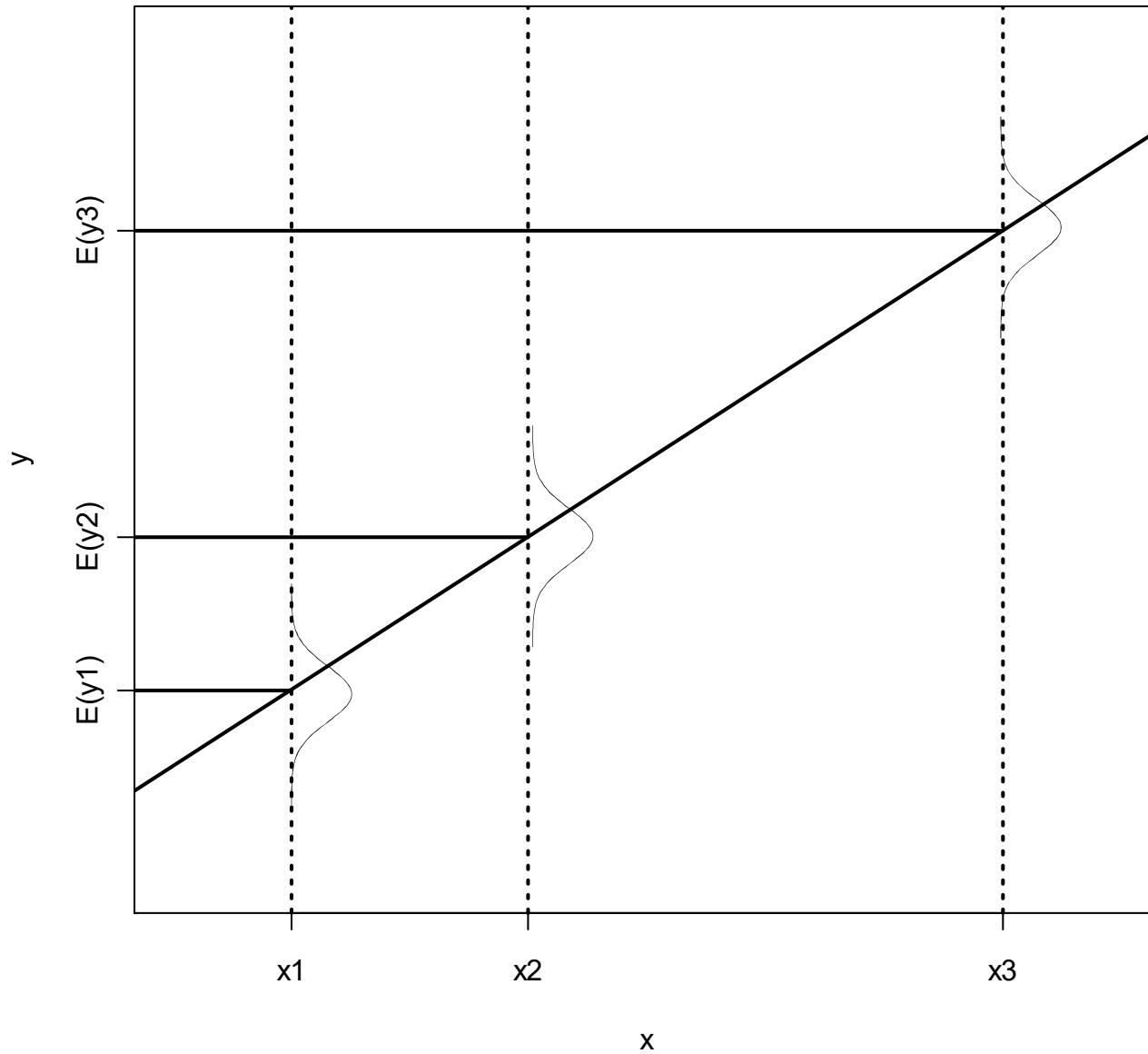
$$E[Y / X] = \beta_0 + \beta_1 \cdot X$$

- **El término constante β_0** es la media esperada de Y cuando X=0, y tiene sentido solo cuando el 0 está dentro del rango de la variable X
- **La pendiente β_1** es el **incremento esperado** en la variable Y cuando X aumenta 1 unidad, y por tanto, β_1 es una **medida del efecto** de X en Y

$$\Delta Y = (\beta_0 + \beta_1 \cdot (X + 1)) - (\beta_0 + \beta_1 \cdot X) = \beta_1$$

- Si la pendiente es 0, $\beta_1=0$ (recta horizontal) quiere decir que **no hay relación lineal** entre las 2 variables, ya que los valores de Y no dependen de X

Regresión lineal simple



Estimación por mínimos cuadrados

- Los parámetros del modelo deben ser estimados a partir de **una muestra**
 - Supongamos n observaciones (x_i, y_i) de las variables X e Y ($i = 1, \dots, n$)
 - El modelo se puede escribir para cada observación

$$y_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i \quad i = 1, \dots, n$$

- **Valores predichos** o ajustados por la recta de regresión. Son las **predicciones** del modelo: medias de la variable Y para los distintos valores de X

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_i \quad i = 1, \dots, n$$

- **Residuos** son las diferencias entre los valores observados y los valores ajustados

$$e_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 \cdot x_i) \quad i = 1, \dots, n$$

Estimación por mínimos cuadrados

- El **método de mínimos cuadrados** estima β_0 y β_1 tal que la suma de los cuadrados de los residuos sean mínimas
- Los **estimadores de mínimos cuadrados** de β_0 y β_1 son los que minimizan

$$\sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 \cdot x_i)^2$$

- Los **estimadores de los coeficientes de regresión** se obtienen igualando a 0 las derivadas parciales respecto a β_0 y β_1

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (y_i - \bar{y}) \cdot (x_i - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \cdot \bar{x}$$

- Los estimadores se resuelven mediante **expresiones algebraicas**
- β_1 está medido en **unidades** que dependen de las unidades de X e Y.
- Son **estimadores de máxima verosimilitud**

Descomposición de la varianza

- Al igual que en el análisis de la varianza, en el análisis de regresión lineal **la variabilidad** se puede descomponer

$$SS_T = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2 = SS_R + SS_E$$

- SS_R varianza explicada** por el modelo de regresión
- SS_E varianza residual o no explicada**

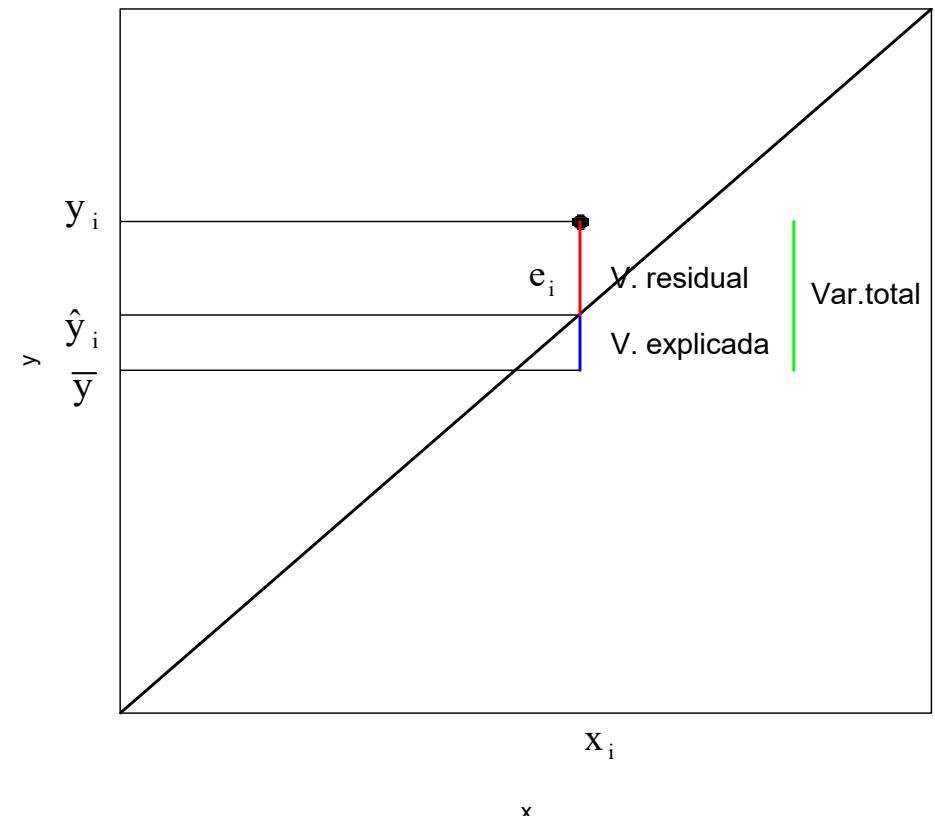


Tabla ANOVA de la regresión lineal simple

- **Tabla del análisis de la varianza de la regresión lineal simple**

Fuente Variación	Suma de cuadrados	gl	Medias cuadráticas
Regresión	$SS_R = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$	1	$MS_R = SS_R / 1$
Residual	$SS_E = \sum_{i=1}^n (y_i - \hat{y}_i)^2$	$n - 2$	$MS_E = SS_E / (n-2)$
Total	$SS_T = \sum_{i=1}^n (y_i - \bar{y})^2$	$n - 1$	$MS_T = SS_T / (n-1)$

- Bajo la hipótesis nula de que el coeficiente β_1 es 0, **el cociente F = MS_R/MS_E** sigue una **distribución F de Snedecor** con (1,n-2) grados de libertad
 - El cociente F es la varianza explicada dividida por la residual: si el modelo es “bueno”, significa que la varianza explicada es “grande” y aceptamos que el modelo lineal **explica una parte significativa de la variabilidad global**

Pruebas de significación de los coeficientes

- Se pueden calcular estimadores de los **errores estándar** $SE(\hat{\beta}_j)$ de los coeficientes de regresión. Además:

$$t_0 = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)} \approx t_{n-2}$$

- Este resultado permite obtener **Intervalos de confianza** al nivel $(1-\alpha)$ de los coeficientes de regresión

$$\hat{\beta}_j \pm t_{n-2,\alpha/2} \cdot SE(\hat{\beta}_j)$$

donde $t_{n-2,\alpha/2}$ es el valor de la distribución t con $n-2$ gl al error $\alpha/2$

- Este resultado se puede utilizar para **contrastar** la significación estadística de cualquier coeficiente
 - $H_0: \beta_j = 0$
 - $H_1: \beta_j \neq 0$

Bondad del ajuste. Coeficiente de determinación

- El **coeficiente de determinación** es una medida del ajuste del modelo y es la proporción de variabilidad de la variable dependiente explicada por el modelo de regresión

$$R^2 = \frac{SS_R}{SS_T} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{\sum e_i^2}{\sum (y_i - \bar{y})^2}$$

- $0 < R^2 < 1$, donde $R^2 = 1$ significa un **ajuste perfecto** y $R^2 = 0$ un ajuste nulo
- En regresión simple $R^2 = r^2$. Tiene más sentido en regresión lineal múltiple
- El coeficiente R^2 se considera que sobrevalora la variabilidad explicada, y por ese motivo se define también el **coeficiente de determinación ajustado** por los grados de libertad para corregir el sobreajuste

$$R_{adj}^2 = 1 - (1 - R^2) \frac{n-1}{n-2} = 1 - \frac{\sum e_i^2 / (n-2)}{\sum (y_i - \bar{y})^2 / (n-1)}$$

Ejemplo: Regresión lineal simple

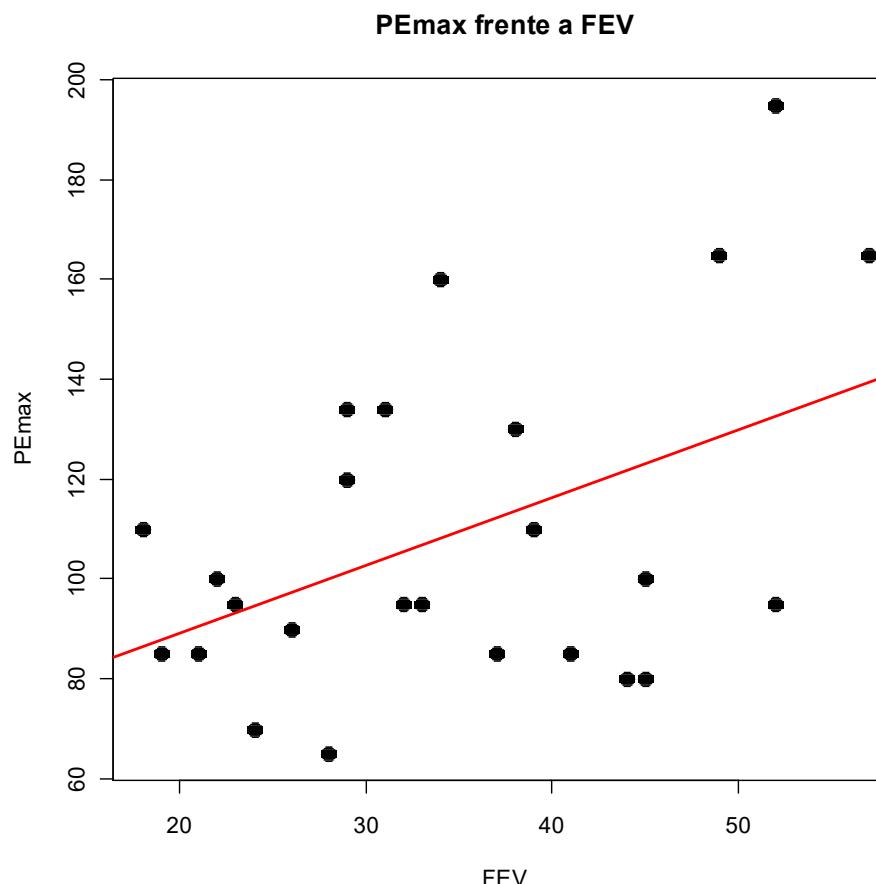
```
> ## Fichero Datos: cystic fibrosis
> xx <- read.csv(file="C://Bioestadistica con R/Datos/cystic fibrosis.csv", sep=";")
> ## Ajuste del modelo de regresión
> lm.fev <- lm( pemax ~ fev , data=xx)
> summary(lm.fev)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  62.114     20.208   3.074  0.00537 **
fev          1.354      0.555   2.439  0.02284 *
Residual standard error: 30.44 on 23 degrees of freedom
Multiple R-squared:  0.2055,    Adjusted R-squared:  0.171
F-statistic: 5.951 on 1 and 23 DF,  p-value: 0.02284
```

- **Modelo estimado:** $\text{PEmax} = 1.35 * \text{fev} + 62.11$
- Por cada aumento en una unidad de la variable “fev”, **esperamos un aumento de 1.35** en la variable PEmax, y este efecto es significativo ($\beta_1 \neq 0$) ($P=0.023$)
- Para el valor $\text{fev}=0$ (si tuviera sentido) el valor esperado de PEmax es 62.11
- La variable fev explica un **20.6% de la variabilidad** de la variable PEmax (**17.1%** ajust.)
- El modelo lineal explica una **parte significativa de la variabilidad global** de la variable PEmax (test F del ANOVA, $P=0.023$). En la regresión simple, el test de la t de Student para la significación del coeficiente β_1 coincide con la prueba global del modelo

Ejemplo: Regresión lineal simple

```
> ## Gráfico de dispersión  
> dev.new()  
> plot ( xx$fev, xx$pemax, pch=16, cex=1.5, main="PEmax frente a FEV",  
+         xlab="FEV", ylab="PEmax" )  
> abline ( lm.fev, col="red", lwd=2 )
```



Ejemplo: Regresión lineal simple

```
> ## Intervalos de confianza
> confint(lm.fev)
      2.5 %    97.5 %
(Intercept) 20.3093964 103.918078
fev          0.2057773  2.501957
> confint(lm.fev) [2,]
      2.5 %    97.5 %
0.2057773 2.5019567
> ## Objetos
> summary(lm.fev)$coeff
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 62.113737 20.2084389 3.073653 0.005372355
fev         1.353867  0.5549926 2.439432 0.022837355
> summary(lm.fev)$coeff[2,1]    ## coeficiente
[1] 1.353867
> summary(lm.fev)$coeff[2,4]    ## p-value
[1] 0.02283735
> paste( round(summary(lm.fev)$coeff[2,1],2) , "(" ,
+        round(confint(lm.fev) [2,1],2) , "-" , round(confint(lm.fev) [2,2],2) , ")" )
[1] "1.35 ( 0.21 - 2.5 )"
```

- Los intervalos de confianza se construyen con la función **confint()**
- Con una confianza del 95%, la variable PEmax se incrementa entre 0.21 y 2.50, por cada unidad de la variable “fev”
- La variable es significativa porque el IC95% no contiene el 0

Ejemplo: Regresión lineal simple

```
> ## Tabla ANOVA
> anova(lm.fev)
Analysis of Variance Table

            Df  Sum Sq Mean Sq F value    Pr(>F)
fev         1  5515.4  5515.4  5.9508 0.02284 *
Residuals 23 21317.2   926.8
---
> ## Valores ajustados y residuos
> fitted(lm.fev)
      1       2       3       4       5       6       7       8
105.43748 87.83721 91.89881 117.62228 132.51482 121.68389 100.02201 86.48334
      9       10      11      12      13      14      15      16
 94.60655 93.25268 114.91455  97.31428 123.03775 123.03775 104.08361 101.37588
      17      18      19      20      21      22      23      24
128.45322 101.37588 113.56068  90.54494 112.20682 108.14522 139.28416 106.79135
      25
132.51482
> resid(lm.fev)
      1       2       3       4       5       6       7
-10.437482 -2.837211  8.101188 -32.622285 -37.514822 -41.683886 -35.022014
      8       9      10      11      12      13      14
 23.516656 -24.606546  1.747321 -4.914551 -7.314280 -23.037753 -43.037753
      15      16      17      18      19      20      21
 29.916385 32.624119 36.546779 18.624119 16.439316 -5.544945 -27.206817
      22      23      24      25
 51.854784 25.715843 -11.791349  62.485178
```

Regresión lineal robusta

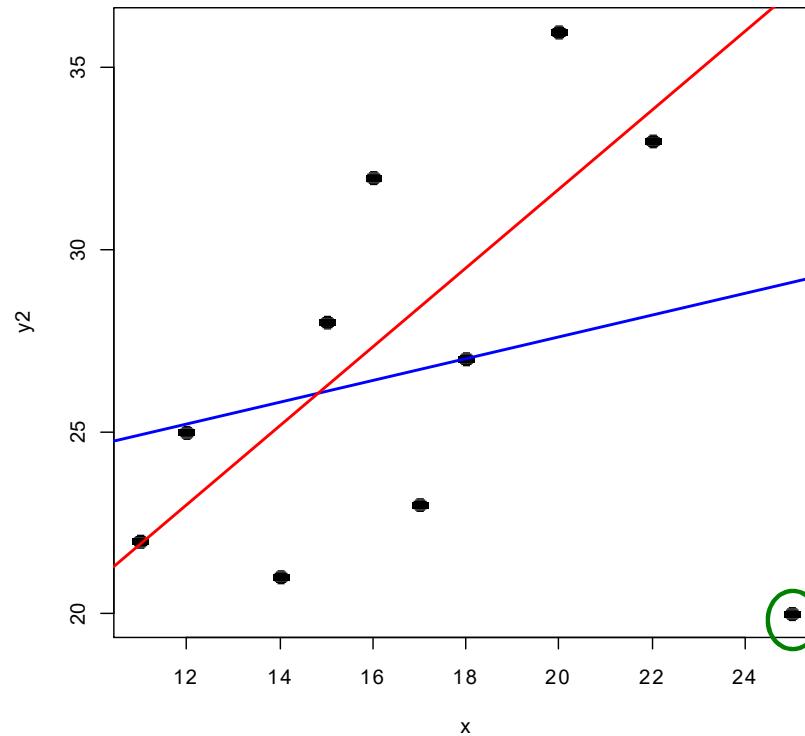
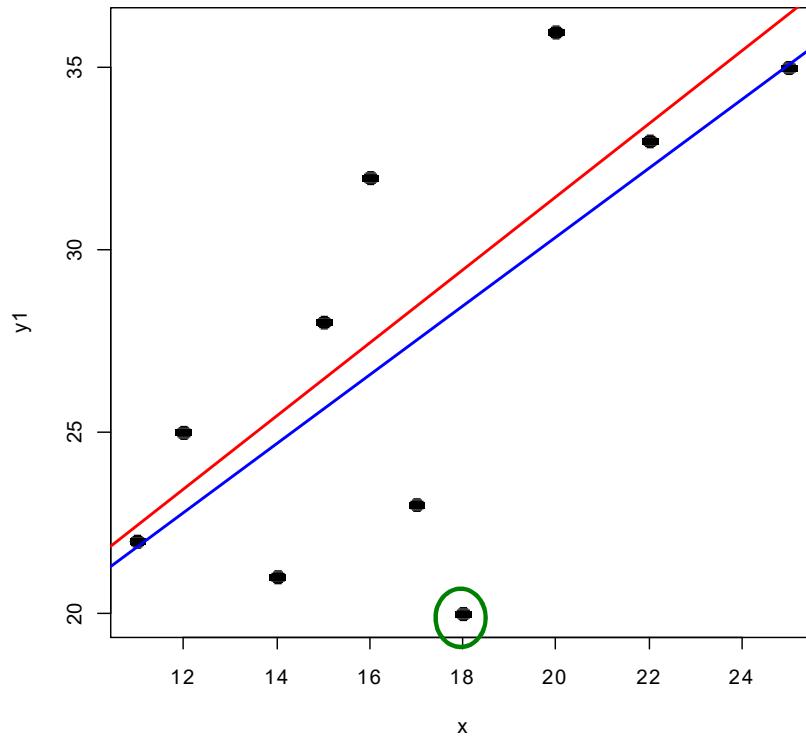
- La regresión lineal basada en el método de mínimos cuadrado puede ser sensible a la **presencia de valores alejados** (outliers)
 - Errores con distribuciones con colas más largas que la distribución normal
 - Muestras pequeñas
- La **regresión robusta** es una alternativa en estos casos
- El **método LTS (least trimmed squares)**
 - Minimiza la suma de los h residuos al cuadrado más pequeños, donde h es una proporción de los datos ($n/2 \leq h \leq n$). Normalmente $h=n/2$, es decir se buscan estimadores que **minimicen el 50% de los residuos más pequeños**

$$\sum_{i=1}^h e_{(i)}^2 \quad e_{(1)}^2 \leq e_{(2)}^2 \leq \dots \leq e_{(n)}^2 \quad \text{residuos al cuadrado ordenados}$$

Ejemplo: Regresión robusta

```
> library(rrcov)
> x <- c(11,12,14,15,16,17,18,20,22,25)
> y1 <- c(22,25,21,28,32,23,20,36,33,35) ## Outlier en Y
> y2 <- c(22,25,21,28,32,23,27,36,33,20) ## Outlier en X
> dev.new(); par(mfrow=c(1,2))
> ## Ejemplo 1
> lm.1 <- lm( y1 ~ x )
> lts.1 <- ltsReg( y1 ~ x , use.correction=F)
> lm.1$coef
(Intercept)          x
11.3793103  0.9482759
> lts.1$coef
Intercept          x
11.401028  1.002571
> plot(x,y1,pch=16,cex=1.5)
> abline(lm.1,col="blue", lwd=2)
> abline(lts.1,col="red", lwd=2)
> ## Ejemplo 2
> lm.2 <- lm( y2 ~ x )
> lts.2 <- ltsReg( y2 ~ x , use.correction=F)
> lm.2$coef
(Intercept)          x
21.6195402  0.2988506
> lts.2$coef
Intercept          x
9.976242  1.084233
> plot(x,y2,pch=16,cex=1.5)
> abline(lm.2,col="blue", lwd=2)
> abline(lts.2,col="red", lwd=2)
```

Ejemplo: Regresión robusta



- Un **outlier en el eje y** “tira” de la recta hacia abajo o hacia arriba. Puede afectar al término constante, pero la pendiente cambia poco
- Un **outlier en el eje x** puede cambiar mucho la pendiente de la recta, y por tanto su efecto puede ser mayor

Ejercicio

- Fichero de datos: cystic fibrosis
 - variable respuesta: pemax
- Analizar la relación de la variable “height” con la variable respuesta “pemax”
 - Gráfico de dispersión
 - Ajustar un modelo de **regresión simple**
 - Interpretación del modelo

Estadística Aplicada a la Investigación Biomédica con R

12 Regresión Lineal Múltiple

- ✓ **Regresión lineal múltiple**
- ✓ **Estimación por mínimos cuadrados**
- ✓ **Tabla ANOVA y pruebas de significación**
- ✓ **Bondad de ajuste**
- ✓ **Supuestos del modelo**
- ✓ **Análisis de influencia**

Regresión lineal múltiple

- Modelo que representa la **relación lineal** de una variable respuesta Y respecto de varias variables predictoras X_1, \dots, X_p
- El **modelo de regresión lineal múltiple** se formula como:

$$Y = \beta_0 + \beta_1 \cdot X_1 + \dots + \beta_p \cdot X_p + \varepsilon$$

- ε es **el error aleatorio**, que se supone sigue una distribución normal con media 0 y varianza desconocida σ^2 que no depende de los valores de las variables predictoras (homocedasticidad)
- $\beta_0, \beta_1, \dots, \beta_p$ son los parámetros del modelo, llamados **coeficientes de regresión**: β_0 es el término constante (intercept) y los coeficientes β_j modelan los **efectos de cada variable**

Regresión lineal múltiple

- El modelo de regresión es **lineal respecto a los parámetros**
- Se pueden analizar **relaciones no lineales** con el modelo de regresión lineal, haciendo **transformaciones** en las variables predictoras
- Ejemplo:

$$Y = \beta_0 + \beta_1 \cdot \ln(X_1) + \beta_2 \cdot X_2$$

- La **regresión polinómica** de 1 variable se ajusta como un modelo de regresión lineal múltiple

$$Y = \beta_0 + \beta_1 \cdot X + \beta_2 \cdot X^2 + \beta_3 \cdot X^3$$

Regresión lineal múltiple

- El modelo de regresión lineal múltiple también se puede expresar con la **media de Y como una función lineal de las variables X_1, \dots, X_p** que representa un plano o **hiperplano**

$$E[Y | X_1, \dots, X_p] = \beta_0 + \beta_1 \cdot X_1 + \dots + \beta_p \cdot X_p$$

- **El término constante β_0** es la media esperada de Y cuando todas las variables predictoras valen 0 ($X_1=0, \dots, X_p=0$), si estos valores tienen sentido
- **El coeficiente de regresión β_j** es el **incremento esperado** en la variable Y cuando X_j aumenta 1 unidad y el resto de variables predictoras permanecen constante. Por tanto, β_j es una **medida del efecto** de X_j en Y

Estimación por mínimos cuadrados

- El **método de mínimos cuadrados** estima los coeficientes β_j ($j=0,1,\dots,p$) tal que la suma de los cuadrados de los residuos sean mínimas
- Los **estimadores de mínimos cuadrados** de los coeficientes β_j ($j=0,1,\dots,p$) son los que minimizan

$$\sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 \cdot x_{i1} - \dots - \hat{\beta}_p \cdot x_{ip})^2$$

- Los **estimadores de los coeficientes de regresión** se obtienen igualando a 0 las derivadas parciales respecto a los coeficientes β_j ($j=0,1,\dots,p$)
 - Los estimadores se resuelven mediante **cálculo matricial**
 - Los coeficientes β_j están medidos en **unidades** que dependen de las unidades de X_j e Y .
 - Son **estimadores de máxima verosimilitud**

Descomposición de la varianza

- Al igual que en el análisis de la varianza, en el análisis de regresión lineal **la variabilidad** se puede descomponer

$$SS_T = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2 = SS_R + SS_E$$

- **SS_R** **varianza explicada** por el modelo de regresión
- **SS_E** **varianza residual** o no explicada
- La **tabla ANOVA** de la regresión lineal múltiple se usa como un **test global** del modelo lineal, de todos los coeficientes de regresión
 - $H_0: \beta_1 = \dots = \beta_p = 0$
 - $H_1: \beta_j \neq 0$ para algún j

Descomposición de la varianza

- Al igual que en el análisis de la varianza, en el análisis de regresión lineal **la variabilidad** se puede descomponer

$$SS_T = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2 = SS_R + SS_E$$

- **SS_R** **varianza explicada** por el modelo de regresión
- **SS_E** **varianza residual** o no explicada
- La **tabla ANOVA** de la regresión lineal múltiple se usa como un **test global** del modelo lineal, de todos los coeficientes de regresión
 - $H_0: \beta_1 = \dots = \beta_p = 0$
 - $H_1: \beta_j \neq 0$ para algún j

Tabla ANOVA de la regresión lineal múltiple

- **Tabla del análisis de la varianza de la regresión lineal múltiple**

Fuente Variación	Suma de cuadrados	gl	Medias cuadráticas
Regresión	$SS_R = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$	p	$MS_R = SS_R / p$
Residual	$SS_E = \sum_{i=1}^n (y_i - \hat{y}_i)^2$	n - p - 1	$MS_E = SS_E / (n-p-1)$
Total	$SS_T = \sum_{i=1}^n (y_i - \bar{y})^2$	n - 1	$MS_T = SS_T / (n-1)$

- Bajo la hipótesis nula $H_0: \beta_1 = \dots = \beta_p = 0$, el cociente $F = MS_R/MS_E$ sigue una **distribución F de Snedecor** con (p, n-p-1) grados de libertad
 - El cociente F es la varianza explicada dividida por la residual: si el modelo es “bueno”, significa que la varianza explicada es “grande” y aceptamos que el modelo lineal **explica una parte significativa de la variabilidad global**

Pruebas de significación de los coeficientes

- Se pueden calcular estimadores de los **errores estándar** $SE(\hat{\beta}_j)$ de los coeficientes de regresión. Además:

$$t_0 = \frac{\hat{\beta}_j}{\hat{SE}(\hat{\beta}_j)} \approx t_{n-p-1}$$

- Este resultado permite obtener **Intervalos de confianza** al nivel $(1-\alpha)$ de los coeficientes de regresión

$$\hat{\beta}_j \pm t_{n-p-1, \alpha/2} \cdot \hat{SE}(\hat{\beta}_j) \quad \text{donde } t_{n-p-1, \alpha/2} \text{ es el valor de la distribución t con } n-p-1 \text{ gl al error } \alpha/2$$

- Este resultado se puede utilizar para **contrastar** cualquier coeficiente
 - $H_0: \beta_j = 0$
 - $H_1: \beta_j \neq 0$
 - Estos contrastes tienen sentido en un determinado modelo con un conjunto de variables predictoras. Por tanto, se considera un test para contrastar el coeficiente β_j de una variable, ajustado por el resto de las variables del modelo

Bondad del ajuste. Coeficiente de determinación

- El **coeficiente de determinación** es una medida del ajuste del modelo y es la proporción de variabilidad de la variable dependiente explicada por el modelo de regresión

$$R^2 = \frac{SS_R}{SS_T} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{\sum e_i^2}{\sum (y_i - \bar{y})^2}$$

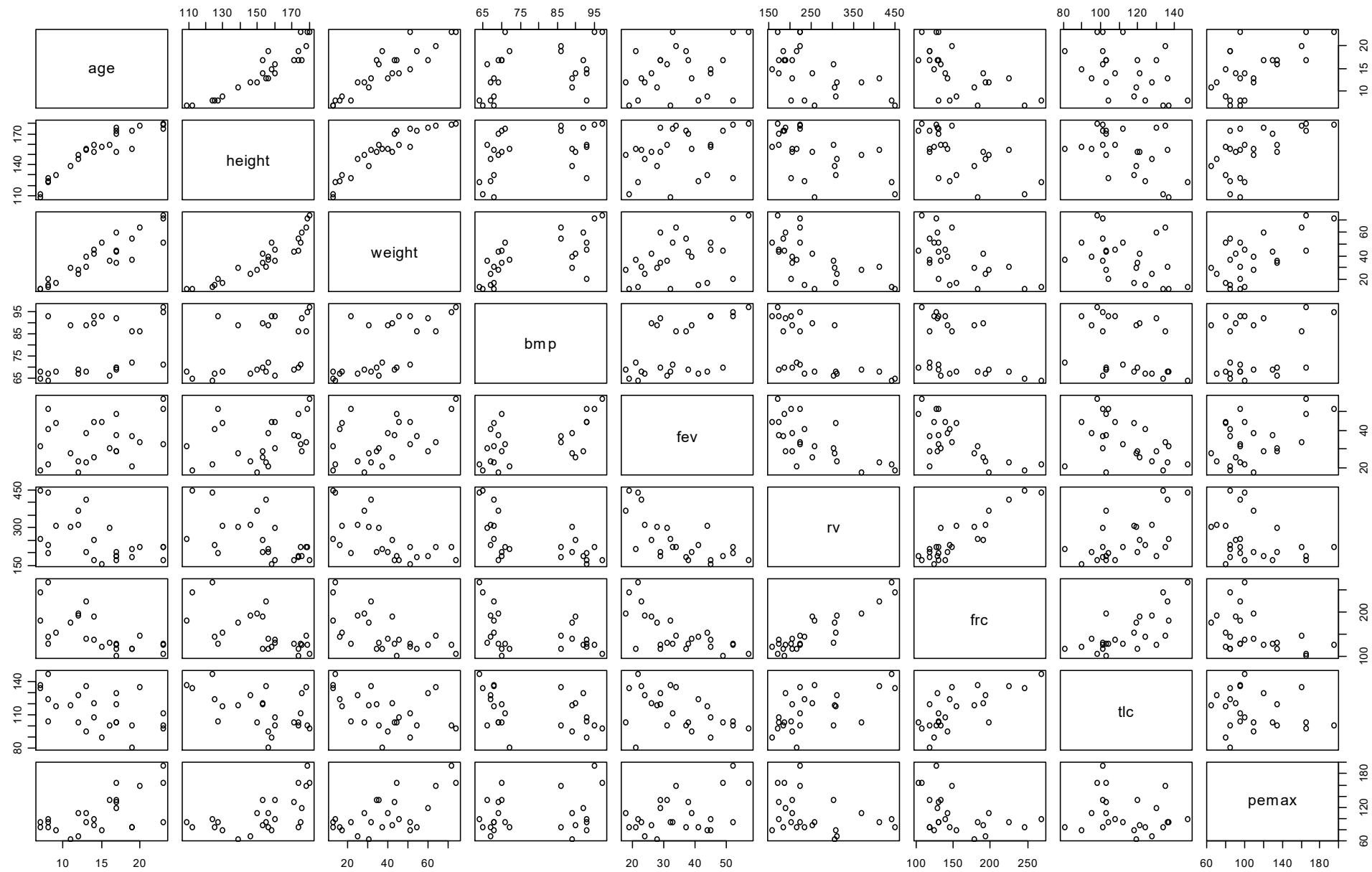
- $0 < R^2 < 1$, donde $R^2 = 1$ significa un **ajuste perfecto** y $R^2 = 0$ un ajuste nulo
- El coeficiente R^2 se considera que sobrevalora la variabilidad explicada, y por ese motivo se define también el **coeficiente de determinación ajustado** por los grados de libertad para corregir el sobreajuste

$$R_{adj}^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1} = 1 - \frac{\sum e_i^2 / (n - p - 1)}{\sum (y_i - \bar{y})^2 / (n - 1)}$$

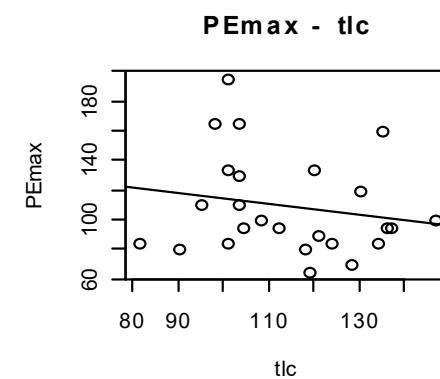
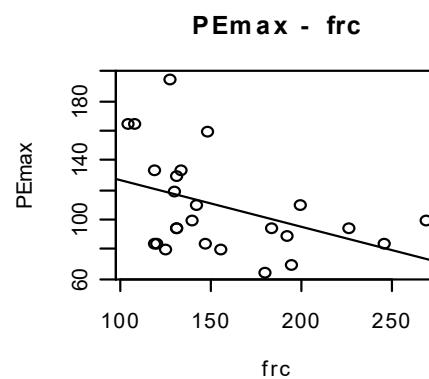
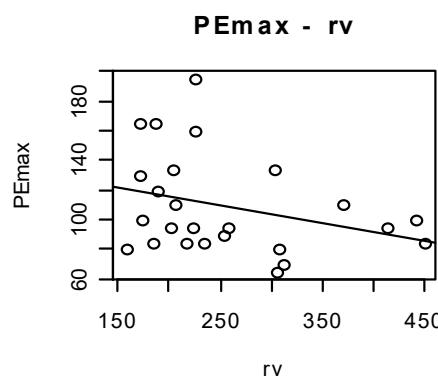
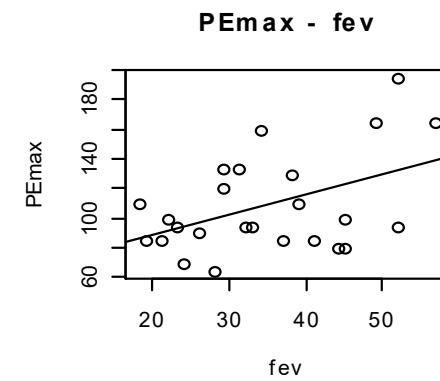
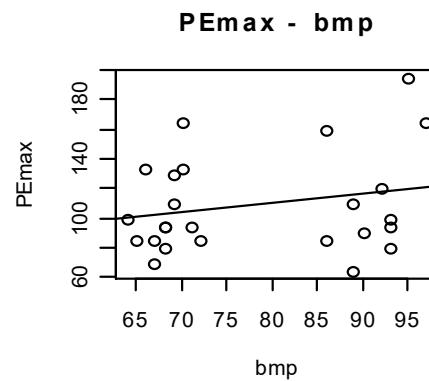
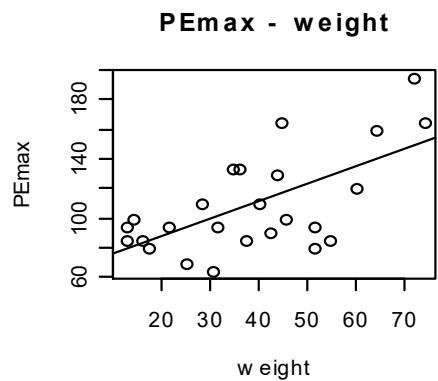
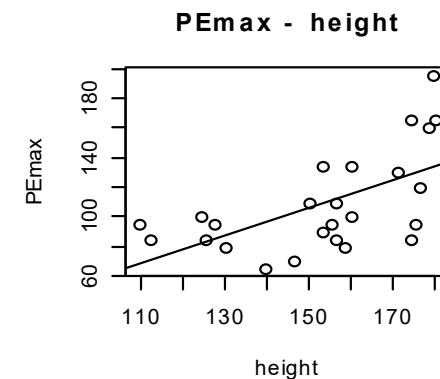
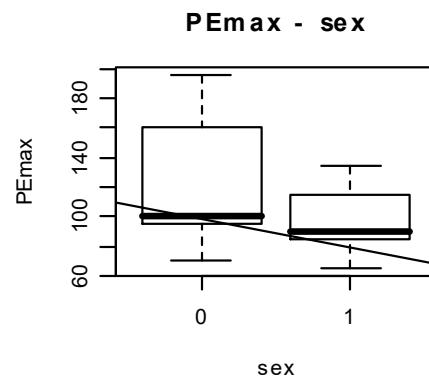
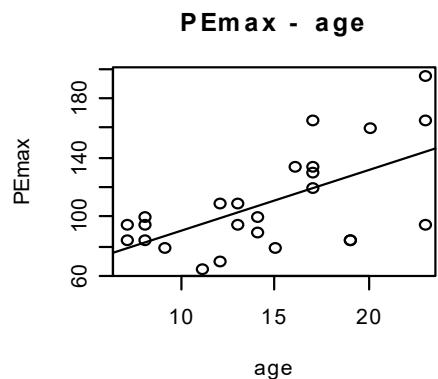
Ejemplo: Regresión lineal múltiple

```
> ## Fichero Datos: cystic fibrosis
> xx <- read.csv(file="C://Bioestadistica con R/Datos/cystic fibrosis.csv", sep=";")
> ## Explorando las covariables
> dev.new()
> pairs ( xx[ , c(2,4:11) ] )
>
> ## Explorando los modelos de regresión simples (variables 2:10)
> dev.new()
> par(mfrow=c(3,3))
>
> for ( i in 2:10 )
+ { lm.w <- lm ( xx$pemax ~ xx[ , i ] )
+   plot ( xx[ , i ], xx$pemax, main=paste ("PEmax - ", names(xx)[i] ) )
+   abline(lm.w)
+   print ( paste ( names(xx)[i] , round( anova(lm.w)$"Pr(>F)"[1], dig=4), sep=" - " ) )
+ }
[1] "age - 0.0011"
[1] "sex - 0.1618"
[1] "height - 0.0015"
[1] "weight - 6e-04"
[1] "bmp - 0.2698"
[1] "fev - 0.0228"
[1] "rv - 0.1244"
[1] "frc - 0.038"
[1] "tlc - 0.3878"
```

Ejemplo: Regresión lineal múltiple



Ejemplo: Regresión lineal múltiple



Ejemplo: Regresión lineal múltiple

```
> lm.mod <- lm( pemax ~ age + fev + rv , data=xx)
> summary(lm.mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -52.44246   48.76422  -1.075  0.29438
age          4.54157    1.19442   3.802  0.00104 **
fev          1.57425    0.60316   2.610  0.01635 *
rv           0.16122    0.08998   1.792  0.08761 .

Residual standard error: 24.51 on 21 degrees of freedom
Multiple R-squared:  0.5297,    Adjusted R-squared:  0.4625
F-statistic: 7.883 on 3 and 21 DF,  p-value: 0.001038
```

- **Modelo estimado:** $\text{PEmax} = 4.54 * \text{age} + 1.57 * \text{fev} + 0.16 * \text{rv} - 52.44$
- El test de ANOVA nos indica que alguno de los 3 coeficientes de regresión es distinto de 0 ($P=0.001$). Los tests de la t de Student indican que son los coeficientes de las variables “age” ($P=0.001$) y “fev” ($P=0.016$), mientras “rv” no es significativa ($P=0.088$)
- Por el aumento en un año en la variable “age”, **esperamos un aumento de 4.54** en la variable PEmax, siendo constantes “fev” y “rv”
- Por el aumento en una unidad de la variable “fev”, **esperamos un aumento de 1.57** en la variable PEmax, siendo constantes “age” y “rv”
- Este modelo explica un **53% de la variabilidad** de la variable PEmax (**46%** ajustado)

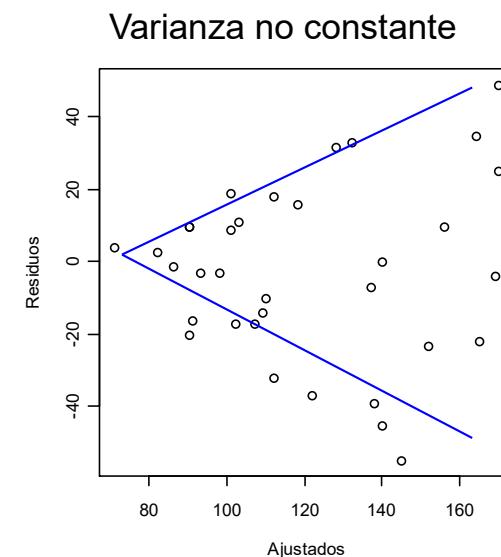
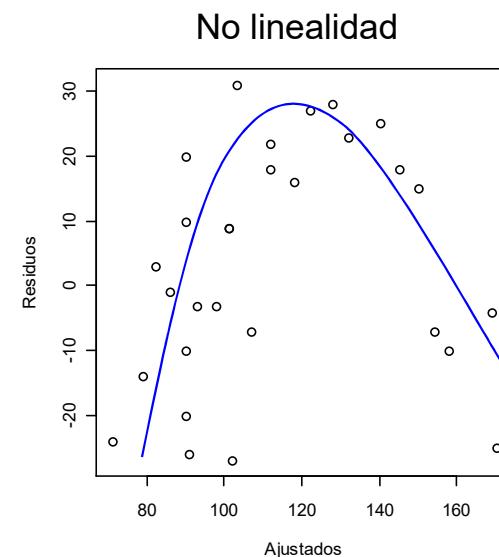
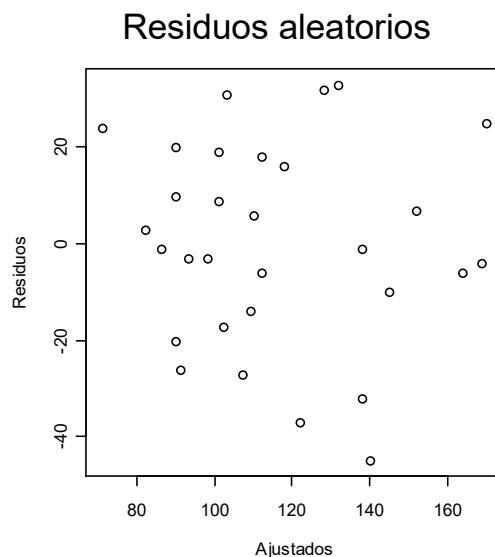
Ejemplo: Regresión lineal múltiple

```
> ## Intervalos de confianza
> confint(lm.mod)
              2.5 %      97.5 %
(Intercept) -153.85321640 48.9682896
age           2.05763389  7.0255134
fev            0.31990822  2.8285832
rv             -0.02590942  0.3483441
>
> w.coef = summary(lm.mod)$coeff
> w.ic   = confint(lm.mod)
>
> for ( i in 2:nrow(w.coef) )
+ {
+   print ( paste( round( w.coef[i,1] , 2 ) , "(" ,
+                 round( w.ic[i,1] , 2 ) , "-" , round(w.ic[i,2] , 2) , ")" ) )
+ }
[1] "4.54 ( 2.06 - 7.03 )"
[1] "1.57 ( 0.32 - 2.83 )"
[1] "0.16 ( -0.03 - 0.35 )"
> ## R2
> summary(lm.mod)$r.squared
[1] 0.5296801
> summary(lm.mod)$adj.r.squared
[1] 0.4624915
```

- Los diferentes objetos que genera la función *lm()* se pueden usar para reportar los datos en el formato que se deseé

Comprobación de los supuestos del modelo

- Los supuestos del modelo pueden ser validados después de ajustar el modelo: **linealidad, normalidad y varianzas constantes**
- **Gráficos de residuos**
 - Gráfico de normalidad de los residuos (QQ plot)
 - Gráfico de los residuos frente a los valores ajustados y gráfico de los residuos frente a las variables predictoras
 - Linealidad y varianzas constantes



Análisis de influencia

- Se llama **observación influyente** a aquella que al quitarla del conjunto de datos, produce un cambio en el ajuste del modelo de regresión lineal
- **Distancia de Cook**
 - Evalúa el **impacto de cada observación** en el ajuste de los coeficientes de regresión estimados
 - Se define para cada una de las observaciones ($k=1, \dots, n$), y estudia las diferencias en los valores ajustados entre el modelo con todas las observaciones y el modelo quitando esa observación

$$D_k = \frac{\sum_{i=1}^n (\hat{y}_i - \hat{y}_{i(k)})^2}{(p+1) \cdot s}$$

$\hat{y}_{i(k)}$ es el valor ajustado para la observación i en el modelo sin la observación k

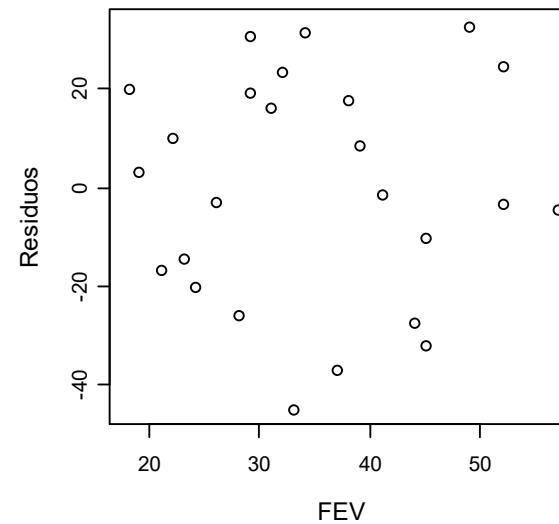
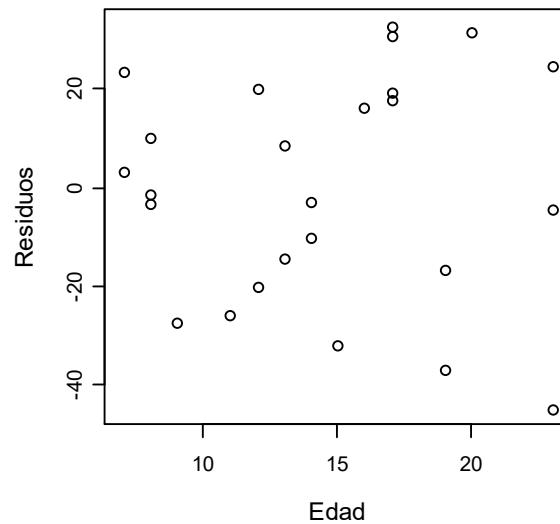
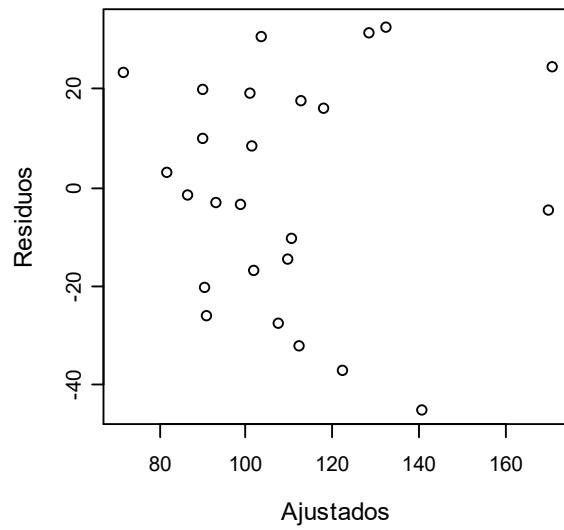
- Se consideran **observaciones influyentes** aquellas con $D_k > 1$
- Otras formulaciones evalúan las diferencia entre coeficientes de regresión

Ejemplo: Supuestos del modelo

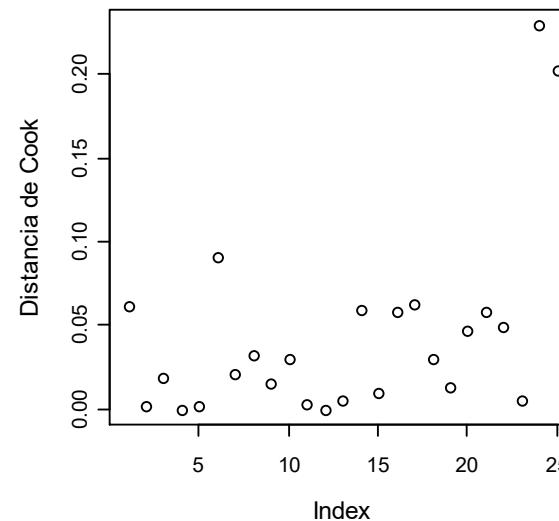
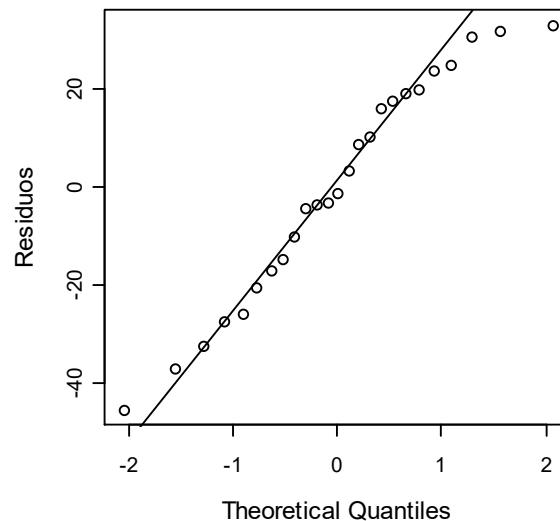
```
> ## Gráfico de los residuos frente a los valores ajustados y los predictores
> dev.new()
> par(mfrow=c(2,3))
> plot(lm.mod$fit, lm.mod$res, xlab="Ajustados", ylab="Residuos", cex.lab=1.2)
> plot(xx$age, lm.mod$res, xlab="Edad", ylab="Residuos", cex.lab=1.2)
> plot(xx$fev, lm.mod$res, xlab="FEV", ylab="Residuos", cex.lab=1.2)
> plot(xx$rv, lm.mod$res, xlab="RV", ylab="Residuos", cex.lab=1.2)
>
> ## QQPlot de los residuos
> qqnorm(lm.mod$res, ylab="Residuos", cex.lab=1.2)
> qqline(lm.mod$res)
>
> ## Distancia de Cook
> cook.mod <- cooks.distance(lm.mod)
> max(cook.mod)
[1] 0.2295093
> which.max(cook.mod)
24
24
> plot(cook.mod, ylab="Distancia de Cook", cex.lab=1.2)
```

- La máxima distancia de Cook de una observación es 0.23 que es totalmente aceptable. Por tanto, no hay ninguna observación que esté influyendo de forma decisiva en el ajuste del modelo de regresión

Ejemplo: Supuestos del modelo

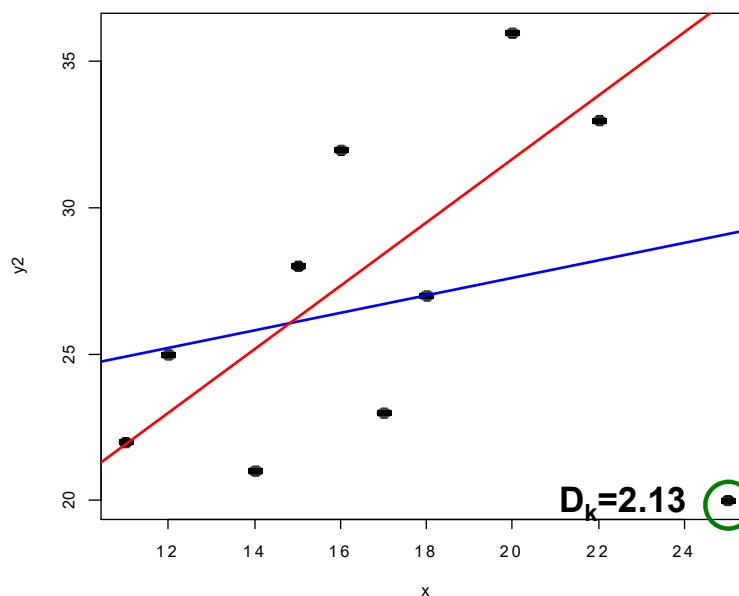
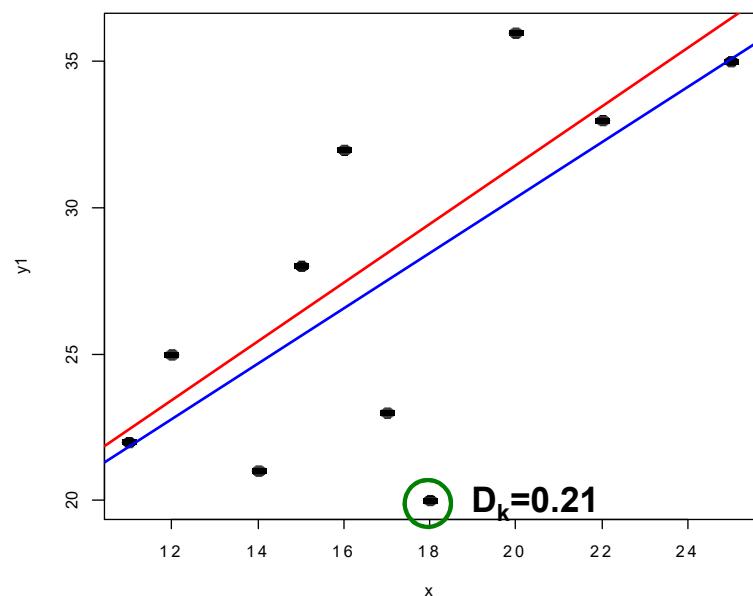


Normal Q-Q Plot



Ejemplo: Distancia de Cook

```
> x <- c(11,12,14,15,16,17,18,20,22,25)
> y1 <- c(22,25,21,28,32,23,20,36,33,35)
> y2 <- c(22,25,21,28,32,23,27,36,33,20)
> lm.1 <- lm( y1 ~ x )
> lm.2 <- lm( y2 ~ x )
> cook.1 <- cooks.distance(lm.1)
> cook.2 <- cooks.distance(lm.2)
> max(cook.1)
[1] 0.2120508
> max(cook.2)
[1] 2.128275
```



Colinealidad

- La **colinealidad** se define como la existencia de una **dependencia lineal** entre algunas variables predictoras del modelo
 - El **coeficiente de correlación de Pearson** se suele utilizar para detectar dependencias lineales entre dos variables predictoras
 - Se suele admitir colinealidad entre dos variables cuando $r > 0.7$ o > 0.8
- En presencia de colinealidad, los estimadores de los coeficientes de regresión obtenidos por el método de mínimos cuadrados son **estimaciones muy inestables**, con errores estándar grandes
 - En el caso de dos predictores muy correlacionados, se aconseja incluir solo una de las dos variables en el modelo

Ejemplo: Colinealidad

```
> cor( xx$height , xx$weight)
[1] 0.9221054
> summary ( lm( pemax ~ height , data=xx ) )
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -33.2757    40.0445  -0.831  0.41453
height       0.9319     0.2596   3.590  0.00155 **
---
> summary ( lm( pemax ~ weight , data=xx ) )
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  63.4493    12.6917   4.999 4.67e-05 ***
weight       1.1880     0.3003   3.956 0.000628 ***
---
> summary ( lm( pemax ~ height + weight , data=xx ) )
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.1420    73.9586   0.664   0.513
height       0.1298     0.6605   0.196   0.846
weight       1.0443     0.7929   1.317   0.201
```

- Al ajustar los 2 modelos de regresión simple, obtenemos que cada variable es significativa
- En el modelo múltiple, los efectos se diluyen ($\beta_1 = 0.13$ y $\beta_2 = 1.04$), y ya no son significativos porque los errores estándar han aumentado

Ejercicio

- Fichero de datos: cystic fibrosis
 - variable respuesta: pemax
- Ajustar el **modelo de regresión múltiple** con las variables predictoras: weight, bmp, fev, rv
 - Interpretar los resultados: tests, coeficientes de regresión
 - Interpretar el coeficiente de determinación
 - Gráfico de los residuos frente a los valores ajustados
 - QQ Plot de los residuos
 - Calcular las distancias de Cook

Estadística Aplicada a la Investigación Biomédica con R

13-14 Regresión Logística

- ✓ **Modelo de Regresión Logística**
- ✓ **Interpretación de los parámetros**
- ✓ **Estimación de los parámetros**
- ✓ **Significación de los coeficientes de regresión**
- ✓ **Variables predictoras categóricas**
- ✓ **Ajuste y calibración del modelo**

Regresión logística binaria

- Modelo de regresión donde la **variable respuesta toma 2 valores**
 - La variable respuesta es una **variable binaria** o dicotómica, y se suelen usar los **códigos 0 y 1**
 - Ejemplos: Presencia o ausencia de una enfermedad, de una complicación, o de un determinado evento clínico de interés. Estudios de casos y controles
 - Se suele usar el código 1 para el evento de interés
- No se puede usar el modelo de regresión lineal
 - Si la variable binaria es modelada como continua como función lineal de los predictores, el modelo predice valores distintos a 0 y 1 y fuera del rango (0,1)
- **Modelo de Regresión Logística**
 - Modela **la probabilidad** de que suceda una de las 2 categorías de la variable respuesta binaria Y

Modelo de regresión logística

- **Modelo de Regresión Logística Simple**
 - Consideramos Y variable respuesta binaria (1/0), y X variable predictora.
La probabilidad de que ocurra el suceso

$$P(Y = 1 / X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

- **Modelo de Regresión Logística Múltiple**
 - Y variable respuesta binaria (1/0). X_1, \dots, X_p variables predictoras

$$P(Y = 1 / X_1, \dots, X_p) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p)}}$$

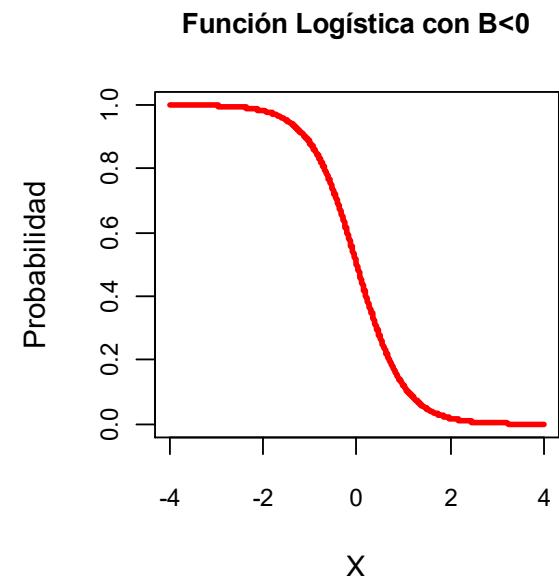
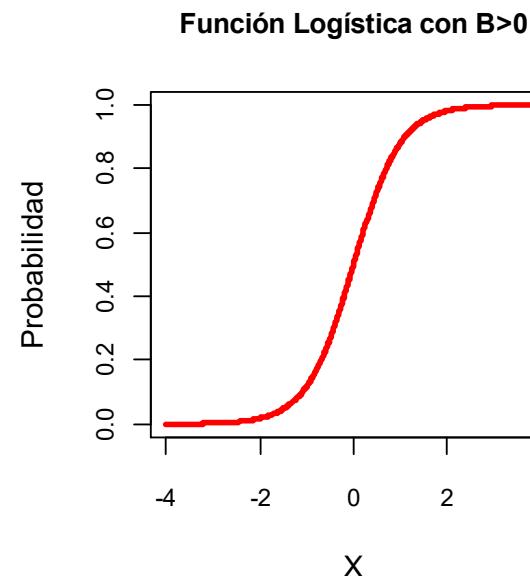
- **Coeficientes de regresión:** $\beta_0, \beta_1, \dots, \beta_p$ parámetros desconocidos
 - **Método de máxima verosimilitud**
 - Estimadores de los coeficientes dan a la muestra observada máxima probabilidad

Función logística

- Modelo de Regresión Logística usa la **función logística** para modelar la probabilidad

$$f(x) = \frac{1}{1 + e^{-B \cdot x}}$$

- Apropiada para modelar una **probabilidad**
 - Tiene forma de S
 - Está entre 0 y 1



- El cambio de $P(Y=1)$ en función de la variable predictora X es lento al principio, a partir de un cierto umbral se hace más rápido y es casi lineal, y finalmente, a partir de otro umbral vuelve a ralentizarse
- Situaciones que se suelen dar dentro del área de la Epidemiología y Biología

La transformación logit

- **Odd** (ventaja)
 - Cociente entre las probabilidades de presentar una característica y no presentarla
 - Cuántas veces más probable es que ocurra un suceso a que no ocurra
 - El odd varía entre 0 e infinito ($0 \leq \text{Prob} \leq 1$)

$$p = P(Y = 1 / X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} \Rightarrow e^{(\beta_0 + \beta_1 X)} = \frac{p}{1 - p} = \text{odds}$$

- **Transformación logit**
 - Tomando logaritmos. Formulación alternativa del modelo de regresión logística
 - El **logit** se modela como una **función lineal** de las variables predictoras
 - El logit toma cualquier valor real

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

Interpretación de los parámetros del modelo

- **Modelo simple**
 - Supongamos dos individuos que toman valores x_1 y x_2 en la variable predictora X. Los odds para esos individuos son:

$$\text{odd}_1 = \frac{p_1}{1-p_1} = e^{(\beta_0 + \beta_1 x_1)} \quad \text{odd}_2 = \frac{p_2}{1-p_2} = e^{(\beta_0 + \beta_1 x_2)}$$

- El **Odds Ratio (OR)**, razón de ventajas, es el cociente entre los 2 odds
 - $\exp(\beta_1(x_1-x_2))$ es el OR entre 2 individuos con valores x_1 y x_2 .

$$\text{OR} = \frac{\text{odd}_1}{\text{odd}_2} = \frac{e^{(\beta_0 + \beta_1 x_1)}}{e^{(\beta_0 + \beta_1 x_2)}} = e^{(\beta_0 + \beta_1 x_1) - (\beta_0 + \beta_1 x_2)} = e^{\beta_1(x_1 - x_2)}$$

- En el caso particular de $x_1=x_2+1$, $\exp(\beta_1)$ es el **OR** entre dos individuos que se diferencian en 1 unidad en términos de la variable predictora continua
- Por este motivo, los predictores binarios se codifican como 0-1 o como 1-2

Interpretación de los parámetros del modelo

- **Odds Ratio** es una medida de asociación que nos indica el número de veces que un individuo con unas características está a más riesgo de sufrir el evento de interés ($Y=1$) que otro con otras características
- Si $\beta_1=0$ significa que X no está asociada a Y
 - $P(Y=1/X)$ sería una constante, tomando el mismo valor para cualquier valor de X , ya que el modelo sería

$$P(Y = 1 / X) = \frac{1}{1 + e^{-(\beta_0 + 0 \cdot X)}} = \frac{1}{1 + e^{-\beta_0}} = \text{cte}$$

- **OR=1** significa que las variables no están asociadas
 - Como $OR=\exp(\beta_1)$, cuando $\beta_1=0$ tendremos que el $OR=\exp(0)=1$
- **Test de asociación** entre X e Y serán equivalentemente:

$$H_0 : \beta_i = 0 \Leftrightarrow H_0 : OR = 1 \Leftrightarrow X_i, Y \text{ no asociadas}$$

$$H_0 : \beta_i \neq 0 \Leftrightarrow H_0 : OR \neq 1 \Leftrightarrow X_i, Y \text{ asociadas}$$

Interpretación de los parámetros del modelo

- **Modelo múltiple**
 - **exp(β_i) es el OR** entre dos individuos que se diferencian solo en X_i en 1 unidad, y son idénticos en todas las demás variables predictoras.
 - **OR de 2 individuos cualesquiera.** Supongamos 2 individuos A y B con valores en las variables predictoras $x_{A1}, x_{A2}, \dots, x_{Ap}$ y $x_{B1}, x_{B2}, \dots, x_{Bp}$ respectivamente

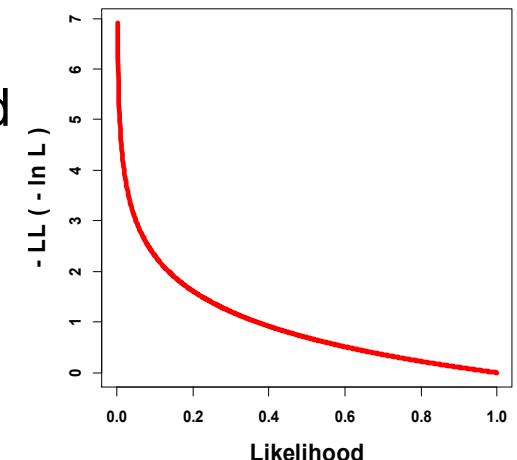
$$OR = \frac{\text{odd}_A}{\text{odd}_B} = \frac{e^{(\beta_0 + \beta_1 x_{A1} + \beta_2 x_{A2} + \dots + \beta_p x_{Ap})}}{e^{(\beta_0 + \beta_1 x_{B1} + \beta_2 x_{B2} + \dots + \beta_p x_{Bp})}} = e^{\beta_1(x_{A1} - x_{B1})} \cdot e^{\beta_2(x_{A2} - x_{B2})} \cdot \dots \cdot e^{\beta_p(x_{Ap} - x_{Bp})}$$

- El modelo de regresión logística es un **modelo multiplicativo**
 - El OR entre 2 individuos con distintos valores en las variables predictoras se puede calcular multiplicando los OR de cada una de las variables

Estimación de los parámetros del modelo

- **Método de máxima verosimilitud**
 - La **función de verosimilitud** $L(\beta)$ de un modelo estimado es la **probabilidad** de reproducir los datos de la muestra a partir del modelo con parámetros β
 - Los estimadores de máxima verosimilitud son los que maximizan $L(\beta)$. Se maximiza el **logaritmo de la verosimilitud**, al que se denota como **LL**
 - Los estimadores $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ dan **máxima probabilidad a la muestra observada**
 - Método iterativo basado en el algoritmo de Newton-Rapshon
- La **bondad de ajuste** se puede evaluar con la verosimilitud
- Se llama **Deviance** (“desvianza”) del modelo a

$$D = -2LL = -2 \cdot \ln L(\text{mod})$$



- Un **modelo es mejor** conforme tiene **menor deviance**
 - El modelo que reproduce los datos perfectamente sería el que tiene $L(\text{mod})=1$ y por tanto $-LL(\text{mod})=0$, ya que $\ln(1)=0$

$$L \in [0,1] \Rightarrow$$

$$LL \in [-\infty, 0] \Rightarrow$$

$$-LL \in [0, \infty]$$

Test del cociente de verosimilitudes

- **Significación global del modelo**

- Compara el logaritmo de la verosimilitud del modelo $L(mod)$ con el logaritmo de la verosimilitud del modelo que solo contiene la constante, $L(0)$
- Test de chi-cuadrado con p g.l., siendo p el número de variables

$$G = -2 \ln LR = -2 \ln \frac{L(0)}{L(mod)} = -2 \cdot \{LL(0) - LL(mod)\} \approx \chi_p^2$$

- **Significación de los coeficientes**

- Compara el logaritmo de la verosimilitud del modelo con todas las variables $L(p)$ con el logaritmo de la verosimilitud del modelo quitando la variable $L(p-1)$
- Test de chi-cuadrado con 1 g.l.

$$G = -2 \ln LR = -2 \ln \frac{L(p-1)}{L(p)} = -2 \cdot \{LL(p-1) - LL(p)\} \approx \chi_1^2$$

- Se puede generalizar para comparar 2 modelos: el modelo con p variables y otro modelo donde se han quitado k variables ($p-k$)

Test de Wald

- Los estimadores de los parámetros $\hat{\beta}_i$, bajo la hipótesis nula $H_0: \beta_i = 0$, se distribuyen según una **distribución normal**, cuando las muestras son suficientemente grandes. Por tanto:

$$\frac{\hat{\beta}_i}{\hat{SE}(\hat{\beta}_i)} \approx N(0,1)$$

- Este resultado se puede usar para comprobar la significación de los coeficientes $\hat{\beta}_i$ y comprobar si X e Y están asociadas
- El **test de Wald** valida la hipótesis nula $H_0: \beta_i = 0$
- Este resultado permite también obtener **Intervalos de confianza** de los coeficientes de regresión

$$\hat{\beta}_i \pm z_\alpha \cdot \hat{SE}(\hat{\beta}_i) \quad \text{donde } z_\alpha \text{ es el valor de la distrib. normal estándar al error } \alpha$$

$$\hat{\beta}_i \pm 1.96 \cdot \hat{SE}(\hat{\beta}_i) \quad \text{es el IC95% en el caso particular de } \alpha=0.05$$

Ejemplo: Variable continua

```
> ## Fichero Datos: Bajo peso al nacer
> xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
> ## Modelos de Regresión Logística Univariantes
> out.1 <- glm ( bajo_pes ~ peso , data=xx, family = binomial )
> summary(out.1)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.99831   0.78529   1.271   0.2036
peso        -0.03099   0.01360  -2.279   0.0227 *
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

> predict ( out.1, data.frame(peso=50), type="response")
 1
0.3655564
> predict ( out.1, data.frame(peso=70), type="response")
 1
0.2366409
```

- El modelo estimado es: $P(\text{bajo_peso}=1/\text{peso}) = \frac{1}{1+e^{-(0.998-0.031\cdot\text{peso})}}$
- $P(\text{bajo peso} / \text{peso madre}=50) = 0.366$ o $P(\text{bajo peso} / \text{peso madre}=70) = 0.237$

Ejemplo: Variable continua

```
> coef.1 <- summary(out.1)$coeff
> coef.1
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) 0.99831432 0.78529087 1.271267 0.20363370
peso        -0.03099284 0.01360147 -2.278639 0.02268856
>
> ## IC95% de los Betas
> confint(out.1)
              2.5 %      97.5 %
(Intercept) -0.48116701 2.611748138
peso        -0.05944039 -0.005842269
> ## OR y IC95%
> exp(coef.1[2,1])
[1] 0.9694825
> exp(coef.1[2,1] - 1.96 * coef.1[2,2])
[1] 0.9439786
> exp(coef.1[2,1] + 1.96 * coef.1[2,2])
[1] 0.9956754
```

- El peso de la madre está asociado al bajo peso al nacer ($P=0.023$, rechazamos $H_0: \beta_1 = 0$)
- El **odds ratio** es 0.969 (IC95%: 0.944 – 0.996). No contiene el 1, está asociado
- Los niños con bajo peso al nacer se presentan un 3.1% veces menos ($1 - 0.969 = 0.031$) en una mujer con un determinado peso al compararla con otra que tenga 1 Kg menos. Es un **efecto protector** (β negativo, $OR < 1$)

Ejemplo: Variable continua

```
> exp( 10*coef.1[2,1] )  
[1] 0.7334995
```

- A veces, es necesario buscar una **escala más adecuada** para la interpretación de una variable continua. Por ejemplo, en nuestro caso 10 Kg
- Los niños con bajo peso al nacer se presentan un 26.7% veces menos ($1 - 0.733 = 0.267$) en una mujer con un determinado peso al compararla con otra que tenga 10 Kg menos
- **Linealidad en el logit**
 - OR=0.733 es válido en **todo el rango** de la variable predictora X, peso de la madre
 - Ese OR es el que se obtiene comparando por ejemplo mujeres de 40 y 50 Kg, y comparando mujeres de 80 y 90 Kg
 - Para **validar este supuesto** se puede categorizar la variable continua (cuartiles) y estudiar la tendencia lineal en los coeficientes

Ejercicios

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
1. Ajustar e interpretar el **modelo de regresión logística** univariante con la variable AGE
 - Calcular IC95% para el odds ratio
 - Calcular la probabilidad de volver a tomar drogas (DFREE=1) para un individuo de 30 años
 - Calcular el OR para un cambio de 5 años de edad

Variables predictoras categóricas

- **Variable predictora binaria**
 - Usar 2 códigos consecutivos: 0-1 o 1-2
 - $\exp(\beta_1)$ se interpreta como el OR entre un individuo que presenta el factor de riesgo ($X=1$) respecto a uno que no lo presenta ($X=0$)
- **Variable predictora categóricas con k categorías ($k>2$)**
 - Códigos arbitrarios, sin significación numérica. No se pueden incluir directamente
 - Definir **k-1 variables dummy** (indicadoras) Z_1, \dots, Z_{k-1}
 - Si el individuo pertenece a la 1º categoría, las K-1 variables dummy valen 0
 - Si el individuo pertenece a la 2º categoría, la variable Z_1 vale 1 y el resto 0
 -
 - Si el individuo pertenece a la Kº categoría, la variable Z_{K-1} vale 1 y el resto 0

Variables predictoras categóricas

- **Variable categórica con k categorías**
 - Por ejemplo, 4 categorías, 3 variables dummy
- **Categoría de referencia**
 - La **categoría de referencia** es la que toma los valores 0 en todas las variables dummy
 - Los coeficientes de las K-1 variables dummy son interpretados como **logit de cada una de las categorías** respecto a la de referencia
 - Se debe elegir la categoría **más protectora** o la **más numerosa**
- **Variable categórica ordinal.** Orden natural en las categorías
 - Variable categórica. Categoría de referencia la primera o última
 - Variable continua. Análisis de tendencia. Un único β evalúa categorías contiguas
 - Se asume que la “distancia” entre las categorías es la misma

Categorías	Z ₁	Z ₂	Z ₃
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1

Ejemplo: Variable binaria

```
> out.2 <- glm(bajo_pes ~ fumador , data=xx, family = binomial )
> summary(out.2)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.0871    0.2147  -5.062 4.14e-07 ***
fumador1     0.7041    0.3196   2.203  0.0276 *
NULL deviance: 234.67 on 188 degrees of freedom
Residual deviance: 229.80 on 187 degrees of freedom
AIC: 233.80

> coef.2 <- summary(out.2)$coeff
> exp( coef.2[2,1] )
[1] 2.021944
> exp( coef.2[2,1] - 1.96 * coef.2[2,2] )
[1] 1.080647
> exp( coef.2[2,1] + 1.96 * coef.2[2,2] )
[1] 3.783154
```

- Fumar está asociado al bajo peso al nacer ($P=0.028$, rechazamos $H_0:\beta_1=0$)
- El **odds ratio** es 2.02 (IC95%: 1.08 – 3.78). No contiene el 1, está asociado
- Los niños con bajo peso al nacer se presentan 2 veces más (el doble) entre las mujeres que fuman con respecto a las que no fuman

Ejemplo: Categoría de referencia

```
> out.2 <- glm(bajo_pes ~ fumador , data=xx, family = binomial )
> summary(out.2)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.0871     0.2147  -5.062 4.14e-07 ***
fumador1     0.7041     0.3196   2.203  0.0276 *

> ## Cambiando la categoría de referencia
> out.2b <- glm(bajo_pes ~ relevel(fumador,"1") , data=xx, family = binomial )
> summary(out.2b)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.3830    0.2368  -1.618   0.1058
relevel(fumador, "1")0 -0.7041    0.3196  -2.203   0.0276 *

> coef.2b <- summary(out.2b)$coeff
> exp( coef.2b[2,1] )
[1] 0.4945736
```

- La categoría de referencia se cambia con la función ***relevel()*** poniendo la categoría entre comillas
- El coeficiente es el mismo, cambiado de signo; y el OR = $\exp(\beta_1) = 0.495 = 1 / 2.022$
- El P-valor del test de Wald es igual

Ejemplo: Variable categórica ($k>2$)

```
> table(xx$raza)

 1  2  3
96 26 67

> prop.table(table(xx$bajo_pes, xx$raza), 2)

      1          2          3
0 0.7604167 0.5769231 0.6268657
1 0.2395833 0.4230769 0.3731343

> out.3 <- glm(bajo_pes ~ raza, data=xx, family = binomial )
> summary(out.3)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.1550     0.2391  -4.830 1.36e-06 ***
raza2        0.8448     0.4634   1.823   0.0683 .
raza3        0.6362     0.3478   1.829   0.0674 .
---
Null deviance: 234.67 on 188 degrees of freedom
Residual deviance: 229.66 on 186 degrees of freedom
AIC: 235.66
```

- Raza “blanca” categoría de referencia: más protectora y más numerosa
- 2 coeficientes, con P-valores no significativos
- R crea las **variables dummy**: “raza2” compara “negra”(2) con “blancas”, y “raza3” “otras razas”(3) con “blancas”

Ejemplo: Variable categórica ($k>2$)

```
> coef.3 <- summary(out.3)$coeff
>
> for ( i in 2:3)
+ { print ( paste ( round( exp( coef.3[i,1] ), dig=2) ,
+                   " (",
+                   round( exp( coef.3[i,1] - 1.96 * coef.3[i,2] ), dig=2) ,
+                   "-",
+                   round( exp( coef.3[i,1] + 1.96 * coef.3[i,2] ), dig=2) ,
+                   ")" , sep ="" ) )
+ }
[1] "2.33 (0.94-5.77)"
[1] "1.89 (0.96-3.74)"
```

- La **interpretación de los Odds Ratio** ($OR=Exp(\beta)$) es la siguiente:
 - Los niños con bajo peso al nacer se presentan 2.33 veces más entre las mujeres de raza negra con respecto a las mujeres de raza blanca (IC95%:0.94-5.77)
 - Los niños con bajo peso al nacer se presentan 1.89 veces más entre las mujeres de otras razas con respecto a las mujeres de raza blanca (IC95%:0.96-3.74)
- Puesto que los OR son de la misma magnitud, se podrían unir las categorías (“negras” y “otras razas”), siempre que tenga sentido

Ejemplo: Variable categórica ($k>2$)

```
## Test del Cociente de verosimilitudes para evaluar la variable globalmente
> out.3$null.deviance
[1] 234.672
> out.3$deviance
[1] 229.6616
> out.3$null.deviance - out.3$deviance
[1] 5.010366
>
> anova(out.3, test="Chisq")
Analysis of Deviance Table
Model: binomial, link: logit
Response: bajo_pes
Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL              188    234.67
raza   2      5.0104     186    229.66  0.08166 .
> drop1(out.3, test="Chisq")
Single term deletions
Df Deviance    AIC    LRT Pr(Chi)
<none>      229.66 235.66
raza       2    234.67 236.67 5.0104 0.08166 .
```

- La variable “raza” no está asociada al bajo peso al nacer ($P=0.082$)

Comprobación del ajuste del modelo

- En regresión lineal
 - R^2 , coeficiente de determinación, proporción de la variabilidad de Y explicada por el modelo
- En regresión logística
 - Coeficientes basados en **la verosimilitud** del modelos
- Coeficiente **R^2 de Cox y Snell**

$$R^2 = 1 - \left[\frac{L(0)}{L(\hat{\beta})} \right]^{2/n}$$

$L(0)$ Verosimilitud del modelo con solo la constante
 $L(\hat{\beta})$ Verosimilitud del modelo considerado

- R^2 así definido no alcanza 1, y por eso se define **R^2 de Nagelkerke**

$$R_C^2 = \frac{R^2}{R_{\max}^2} \quad R_{\max}^2 = 1 - [L(0)]^{2/n} \quad \text{cuando } L(\hat{\beta}) = 1 \text{ (max.prob.)}$$

Ejemplo: Ajuste del modelo

```
LogisticModelFit <- function ( mod )
{
  ## tamaño muestral
  n <- mod$df.null+1

  LLB <- - ( mod$deviance /2 )
  LL0 <- - ( mod$null.deviance /2 )
  L0 <- exp (LL0)
  LB <- exp (LLB)

  R2Cox <- 1 - (L0/LB)**(2/n)
  R2max <- 1 - L0** (2/n)
  R2Nag <- R2Cox /R2max

  list ( R2Cox=R2Cox, R2Nag=R2Nag )
}

> out.w <- glm( bajo_pes ~ edad + peso + raza + fumador + hta + irr_urin + part_pre2
+                   , data=xx, family = binomial )
> LogisticModelFit (out.w)
$R2Cox
[1] 0.1814351

$R2Nag
[1] 0.2551496
```

Calibración del modelo

- **Calibración del modelo**

- Evaluar la concordancia entre las probabilidades observadas (p_i) y las predichas por el modelo (π_i).

$$\sum_{i=1}^n \pi_i = \sum_{i=1}^n p_i = n_1$$

donde n_1 es el número de obs. con $Y=1$ (eventos)

- **Test de bondad de ajuste de Hosmer y Lemeshow**

- Se ordenan los n individuos según las predicciones π_i y se dividen en **los deciles de riesgo** ($g=10$ grupos del mismo tamaño)
 - Para muestras pequeñas, se puede tomar g menor de 10
 - El test compara las probabilidades predichas y observadas en estos g grupos

$$\chi^2 = \sum_{j=1}^2 \sum_{i=1}^g \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \approx \chi^2_{g-2}$$

Ejemplo: Test de Hosmer-Lemeshow

```
> ## Test de Hosmer-Lemeshow
> library(ResourceSelection)
>
> hoslem.test( xx$bajo_pes , fitted(out.w) , g=10 )

Hosmer and Lemeshow goodness of fit (GOF) test

data: xx$bajo_pes, fitted(out.w)
X-squared = 4.6762, df = 8, p-value = 0.7916
```

- El test está en la librería ***ResourceSelection***
- La prueba de Hosmer-Lemeshow no es significativa ($P=0.792$) lo que significa que el modelo está bien calibrado, hay concordancia entre lo observado y lo predicho

Ejercicios

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
1. Ajustar e interpretar el **modelo de regresión logística** univariante con la variable TREAT
 - Calcular IC95% para el odds ratio
 2. Ajustar e interpretar el **modelo de regresión logística** univariante con la variable IVHX (3 categorías)
 - Calcular IC95% para los 2 odds ratios
 - Calcular la significación global de la variable (función *anova()*)

Estadística Aplicada a la Investigación Biomédica con R

15 Variables de confusión e Interacciones

- ✓ **Variables de confusión**
- ✓ **Interacciones entre variables predictoras**

Variables de confusión

- Se desea analizar la relación entre la variable predictora X y la variable respuesta Y, pero hay una variable Z que puede **afectar** a la relación
- Z es una **variable de confusión o de control** en la asociación entre X e Y cuando
 - Z está relacionada con ambas, con X y con Y
 - Z no se encuentra en el camino causal entre X e Y (es externa a la relación)
- Ejemplos de variables de confusión en estudios epidemiológicos
 - edad, sexo, hospital o lugar de procedencia, ...
 - **factores de riesgo conocidos** para el suceso que estemos estudiando
- La **confusión** se presenta especialmente en estudios observacionales de investigación **no experimental**
 - No se controla los niveles de “exposición” de los individuos (no aleatorización)

Variables de confusión

- Se pretende decidir si una determinada variable Z es de confusión en la relación entre X e Y, en la que estamos interesados
 - Ajustamos el modelo de regresión entre X e Y

X \longrightarrow Y siendo B_{1X} el coeficiente de regresión de X

- Ajustamos el modelo de regresión entre X, Z e Y

X, Z \longrightarrow Y siendo B_{2X} el coeficiente de regresión de X

- Si se ha producido un cambio en el coeficiente de regresión de X, entre **un 15% y un 25%**, diremos que Z está **confundiendo la relación** entre X e Y
 - La variable Z tiene que ser **incluida en el modelo**, independientemente de su significación estadística
 - No es muy importante comprobar que la variable Z esté relacionada con X y con Y

Modelos de regresión múltiple

- En los modelos de regresión múltiples, **los coeficientes de regresión** quedan ajustados por la influencia que el resto de las variables tienen con la variable respuesta, y por las relaciones que existen entre las variables predictoras
 - Las variables de confusión o control producen **sesgos** en la estimación de la magnitud de los efectos
 - Este sesgo se elimina a través de análisis estratificados o del ajuste de modelos multivariantes
- Los modelos de regresión múltiples son métodos más eficientes porque permiten estudiar efectos de varias variables **simultáneamente**
- Los modelos multivariantes de regresión logística deberían contener como máximo **1 variable por cada 10 individuos** de la categoría menos numerosa de la variable binaria dependiente

Ejemplo: Variable de confusión

```
> ## Fichero Datos: Bajo peso al nacer
> xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
> ## Variables de confusión
> out.2 <- glm(bajo_pes ~ fumador , data=xx, family = binomial )
> summary(out.2)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.0871     0.2147  -5.062 4.14e-07 ***
fumador1      0.7041     0.3196   2.203  0.0276 *
                                          
> out.4 <- glm(bajo_pes ~ fumador + raza , data=xx, family = binomial )
> summary(out.4)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.8405     0.3529  -5.216 1.83e-07 ***
fumador1      1.1160     0.3692   3.023  0.00251 **
raza2         1.0841     0.4900   2.212  0.02693 *
raza3         1.1086     0.4003   2.769  0.00562 **
                                          
> abs(1.116 - 0.704 ) / 0.704
[1] 0.5852273
```

- La variable “raza” está actuando como **variable de confusión** en la relación entre “fumador” y bajo peso al nacer
 - El coeficiente de regresión ha incrementado un 58.5%
 - “raza” se debe incluir en el modelo, independiente de su significación estadística

Ejemplo: Variable de confusión

```
> coef.4 <- summary(out.4)$coeff
>
> for ( i in 2:4)
+ { print ( paste ( row.names(coef.4)[i], " " ,
+                 round( exp( coef.4[i,1] ), dig=2) ,
+                 " (",
+                 round( exp( coef.4[i,1] - 1.96 * coef.4[i,2] ), dig=2) ,
+                 "-",
+                 round( exp( coef.4[i,1] + 1.96 * coef.4[i,2] ), dig=2) ,
+                 ")" , sep ="" ) )
+ }
[1] "fumador1  3.05 (1.48-6.29)"
[1] "raza2    2.96 (1.13-7.72)"
[1] "raza3    3.03 (1.38-6.64)"
```

- El **OR** de las mujeres fumadoras respecto a las no fumadoras es ahora bastante mayor, **3.05** (IC95%: 1.48 – 6.29) que el que se había estimado antes (2.02)
- Los niños con bajo peso al nacer se presentan 3 veces más (el triple) entre las mujeres que fuman con respecto a las mujeres que no fuman, ajustado por raza
- Los ORs de raza son ahora aproximadamente 3 y son estadísticamente significativos ya que los IC95% no contienen el valor 1

Ejemplo: Variable de confusión

```
## Relación entre la variable predictora y la variable de confusión
> table(xx$raza, xx$fumador)
  0  1
1 44 52
2 16 10
3 55 12

> prop.table(table(xx$raza, xx$fumador), 1)
      0          1
1 0.4583333 0.5416667
2 0.6153846 0.3846154
3 0.8208955 0.1791045

> chisq.test(table(xx$raza, xx$fumador))

Pearson's Chi-squared test

data: table(xx$raza, xx$fumador)
X-squared = 21.779, df = 2, p-value = 1.865e-05
```

- La variable de confusión “raza” está relacionada con el factor de interés “fumador”
 - Los porcentajes de fumadoras en cada raza son distintos
 - Esa diferencia es significativa ($P<0.001$), evaluado con el test de chi-cuadrado

Interacción entre variables predictoras

- La **interacción** se produce cuando la magnitud del efecto de una determinada “exposición” sobre la respuesta varía según los valores de otras variables, llamadas variables modificadoras
 - Las variables modificadoras pueden cambiar **el sentido y la intensidad** de la relación entre la variable de estudio y la variable predictora
 - La interacción puede darse en cualquier tipo de diseño
- Existe una **interacción** entre las variables X y Z cuando la relación entre la variable predictora X y la variable respuesta Y es diferente en los niveles de una tercera variable Z
 - Si Z es una **variable de confusión**, el efecto de X es igual en todos los niveles de Z, aunque para calcularlo, debemos incluir a Z también en el modelo
 - Al menos una de las variables de una interacción debe ser un **factor de riesgo**
- Se dice que Z es una **variable modificadora** del efecto de X en Y

Interacción entre variables predictoras

- El **modelo de interacción** entre X y Z se formula añadiendo un **término producto** de las 2 variables al modelo de efectos principales
 - **Principio jerárquico:** el modelo debe contener los términos de orden inferior

$$\log it(p) = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 X * Z$$

- **Interpretación de una interacción**
 - Supongamos que X y Z son variables binarias codificadas 0, 1
 - $\exp(\beta_1)$ es el efecto de X en Y cuando Z=0
 - $\exp(\beta_1 + \beta_3)$ es el efecto de X en Y cuando Z=1
 - El signo y la magnitud de β_3 determinan la interpretación de la interacción
- **Evaluación de la interacción** con el test $H_0: \beta_3=0$
 - Si $\beta_3=0$ el efecto de X en Y es igual en todos los niveles de Z (β_1)

Interacción entre variables predictoras

- En la práctica, para interpretar una interacción con variables que no son binarias, se fijan **niveles de Z**
- Se calcula **el OR** de un individuo con valor $X = x_1$ con respecto a otro con valor $X = x_2$ en un valor de $Z = z$

$$OR(X = x_1 / X = x_2)_{Z=z} = \exp(\beta_1(x_1 - x_2) + \beta_3 \cdot z \cdot (x_1 - x_2))$$

- Con intervalos de confianza:

$$OR(X = x_1 / X = x_2)_{Z=z} \pm z_{1-\alpha/2} \times \sqrt{((x_1 - x_2) \cdot \text{Var}(\hat{\beta}_1) + [z \cdot (x_1 - x_2)]^2 + 2 \cdot z \cdot (x_1 - x_2) \cdot \text{Cov}(\hat{\beta}_1, \hat{\beta}_3))}$$

Ejemplo: Interacción

```
> ## Crear la variable peso dicotómica
> xx$peso_gr <- NA
> xx$peso_gr [ xx$peso < 49.5 ] <- 1
> xx$peso_gr [ xx$peso >= 49.5 ] <- 0
>
> ## Modelo con interacción
> out.5.int <- glm(bajo_pes ~ peso_gr * edad , data=xx, family = binomial )
> summary(out.5.int)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.77450   0.91010   0.851   0.3948
peso_gr      -1.94409   1.72481  -1.127   0.2597
edad        -0.07957   0.03963  -2.008   0.0447 *
peso_gr:edad  0.13220   0.07570   1.746   0.0807 .
```

- Consideramos peso como variable binaria (pto. corte = 49.5)
- La **interacción** entre edad y peso es estadísticamente significativa con P=0.081, si tomamos $\alpha=0.1$
 - Al introducir una interacción, las variables pueden dejar de ser significativas
- El **efecto que el peso de la madre tiene en el bajo peso de los niños al nacer depende de la edad de la madre**

Ejemplo: Interacción

```
> ## Test del cociente de verosimilitudes para evaluar la interacción
> out.5.main <- glm(bajo_pes ~ peso_gr + edad , data=xx, family = binomial )
> summary(out.5.main)
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.02689    0.76215 -0.035  0.97185
peso_gr       1.01012    0.36426  2.773  0.00555 **
edad        -0.04423    0.03222 -1.373  0.16987
>
> anova(out.5.int, out.5.main, test="Chisq")
Model 1: bajo_pes ~ peso_gr * edad
Model 2: bajo_pes ~ peso_gr + edad
Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1          185     221.14
2          186     224.29 -1   -3.1468   0.07607 .
---
---
```

- El test del cociente de verosimilitudes comparando el modelo con interacción y el modelo de efectos principales, nos indica que la interacción es significativa ($P=0.076$), si tomamos $\alpha=0.1$

Ejemplo: Interacción

```
> ## Coeficientes de Regresión
> coef.5 <- summary(out.5.int)$coeff
> B1<-coef.5[2,1] # peso_gr
> B2<-coef.5[3,1] # edad
> B3<-coef.5[4,1] # Interacción
>
> ## Matriz de covarianzas entre los coeficientes
> cov.est <- summary(out.5.int)$cov.unscaled
> cov.est
            (Intercept)      peso_gr          edad  peso_gr:edad
(Intercept)   0.82827928 -0.82827928 -0.035266546  0.035266546
peso_gr       -0.82827928  2.97495564  0.035266546 -0.127603824
edad         -0.03526655  0.03526655  0.001570894 -0.001570894
peso_gr:edad  0.03526655 -0.12760382 -0.001570894  0.005730240
> var_B1<-cov.est[2,2]
> var_B3<-cov.est[4,4]
> cov_B13<-cov.est[2,4]
>
> ## Cálculo de los ORs del peso de la madre para los distintos niveles de edad
> edad_level <- 15:25
> OR    <- rep(NA,length(edad_level))
> OR_L  <- rep(NA,length(edad_level))
> OR_U  <- rep(NA,length(edad_level))
```

- Se extraen los valores de las varianzas y covarianzas de los estimadores de regresión
- Se definen los niveles de edad en los que se está interesado: 15 - 25

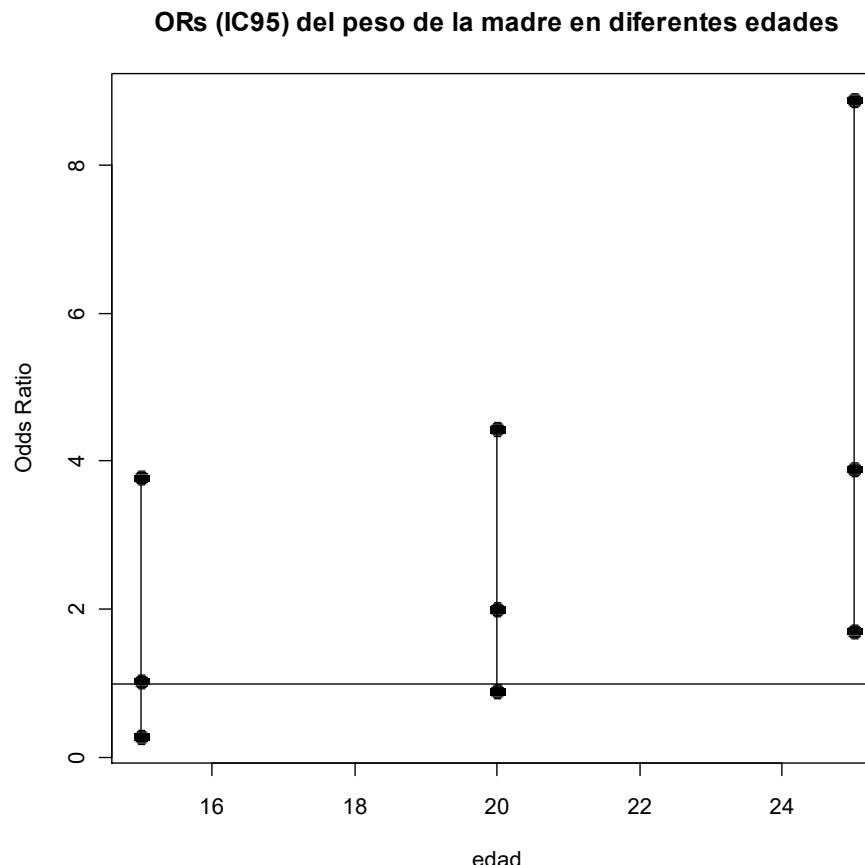
Ejemplo: Interacción

```
> i <- 0
> for (x in edad_level)
+ {
+   i <- i+1
+   SE <- sqrt(var_B1 + x^2 * var_B3 + 2*x *cov_B13)
+
+   OR[i]   <- exp( B1+x*B3)
+   OR_L[i] <- exp( (B1+x*B3) - 1.96 * SE  )
+   OR_U[i] <- exp( (B1+x*B3) + 1.96 * SE  )
+
+   print ( paste ( "Edad ", edad_level[i] , " OR = ",
+                 round( OR[i], dig=2) ,      " (",
+                 round( OR_L[i] , dig=2) , "-",
+                 round( OR_U[i] , dig=2) , ")" , sep ="" ) )
+ }
[1] "Edad 15 OR = 1.04 (0.28-3.79)"
      . . .
[1] "Edad 20 OR = 2.01 (0.91-4.44)"
      . . .
[1] "Edad 25 OR = 3.9 (1.71-8.88)"
```

- El riesgo que tienen las mujeres de menos de 49.5kg, de peso de tener un niño con bajo peso al nacer va aumentando con la edad, siendo el OR=2 en mujeres con 20 años, y OR=3.9 en las mujeres con 25 años
- Para mujeres con 15 años, el peso de la madre no es una factor de riesgo

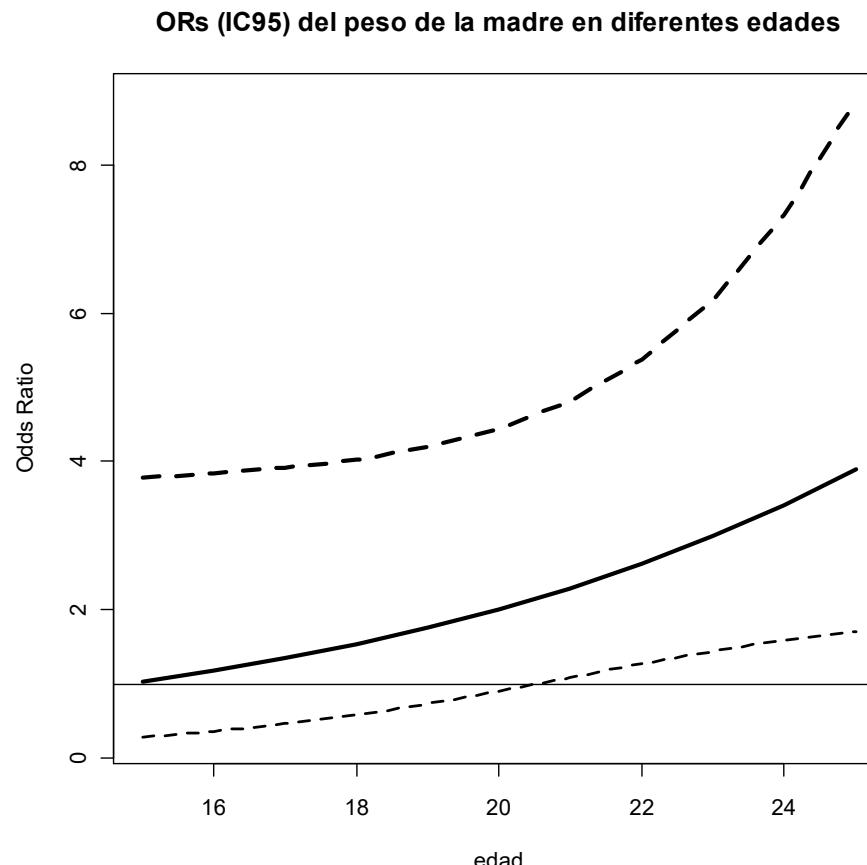
Ejemplo: Interacción

```
> ind <- c(1,6,11) ## elementos de los ORs para dibujar
> dev.new()
> plot ( c(edad_level[ind], edad_level[ind], edad_level[ind]),
+         c(OR[ind], OR_L[ind], OR_U[ind]),
+         main="ORs (IC95) del peso de la madre en diferentes edades",
+         xlab="edad", ylab="Odds Ratio", pch=16, cex= 1.5)
> for ( i in ind ) {segments( edad_level[i], OR_L[i], edad_level[i], OR_U[i])}
> abline(h=1)
```



Ejemplo: Interacción

```
> dev.new()
> plot ( edad_level,OR, type="l", ylim=c(min(OR,OR_L,OR_U),max(OR,OR_L,OR_U) ),
+         main="ORs (IC95) del peso de la madre en diferentes edades",
+         xlab="edad", ylab="Odds Ratio", lwd=3, lty=1)
> lines ( edad_level, OR_L, lwd=2, lty=2)
> lines ( edad_level, OR_U, lwd=3, lty=2)
> abline(h=1)
```



Ejemplo: Interacción 2 variables categóricas

```
> out.6 <- glm ( bajo_pes ~ fumador * peso_gr , data=xx, family = binomial )
> summary(out.6)
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.5106     0.2680 -5.637 1.73e-08 ***
fumador1      1.0894     0.3882  2.806 0.005010 **
peso_gr        1.7983     0.5160  3.485 0.000492 ***
fumador1:peso_gr -1.6647     0.7346 -2.266 0.023431 *
> ## Creación de una variable uniendo las 2 categóricas
> xx$peso_fuma <- NA
> xx$peso_fuma [ xx$fumador == 0 & xx$peso_gr == 0 ] = 0
> xx$peso_fuma [ xx$fumador == 0 & xx$peso_gr == 1 ] = 1
> xx$peso_fuma [ xx$fumador == 1 & xx$peso_gr == 0 ] = 2
> xx$peso_fuma [ xx$fumador == 1 & xx$peso_gr == 1 ] = 3
> xx$peso_fuma <- factor(xx$peso_fuma)
>
> out.7 <- glm ( bajo_pes ~ peso_fuma , data=xx, family = binomial )
> summary(out.7)
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.5106     0.2680 -5.637 1.73e-08 ***
peso_fuma1    1.7983     0.5160  3.485 0.000492 ***
peso_fuma2    1.0894     0.3882  2.806 0.005010 **
peso_fuma3    1.2229     0.5160  2.370 0.017789 *
> ## ORs
> exp( summary(out.7)$coeff[ 2:4, 1] )
peso_fuma1 peso_fuma2 peso_fuma3
 6.039216   2.972426   3.397059
```

Ejercicios

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
1. Ajustar e interpretar el modelo de **regresión logística** multivariante con la variable TREAT y RACE para evaluar si RACE es una **variable de confusión** para el efecto de TREAT
 - Previamente, hay que ajustar el modelo univariante con TREAT
 2. Ajustar e interpretar el modelo de **regresión logística** con **la interacción** entre las variables RACE y SITE

Estadística Aplicada a la Investigación Biomédica con R

16 Construcción de un modelo de Regresión Logística

- ✓ **Estrategia para la construcción de un modelo**
- ✓ **Tratamiento de variables continuas**
- ✓ **Análisis de influencia**

Selección de variables. Modelo final

- **Estrategia general** para la selección de las variables que van a formar parte del modelo final
 - Estrategia para aplicar cuando el número de variables predictoras es moderado
 - Un modelo que **explique** bien los datos
 - Un modelo simple de **interpretar**
- **Principio de parsimonia**
 - El modelo debe incluir el mínimo número de variables posible que logren explicar adecuadamente el fenómeno en estudio
- El modelo debe incluir:
 - las **variables clínicas relevantes**
 - los **factores de riesgos conocidos**
 - las variables de confusión y control

Estrategia para la construcción del modelo

Paso 1: Análisis Univariante

- **Regresión Logística Simple** (ORs crudos, sin ajustar)
- Alternativamente, test de chi-cuadrado para las variables categóricas o t-Student para las continuas

Paso 2: Modelo multivariante completo, que incluye

- Todas las variables con **P<0.25** en el análisis univariante (Paso 1)
- Todas las **variables clínicamente** importantes

Estrategia para la construcción del modelo

Paso 3: Modelo multivariante preliminar de efectos principales, de una **forma iterativa** hay que probar con varios modelos derivados del modelo completo

Las variables menos significativas se van quitando del modelo completo, evaluando si son variables de confusión

Finalmente. este modelo incluirá:

- Variables que son **estadísticamente significativas**
- **Variables de confusión**, evaluadas al comparar los coeficientes de regresión del resto de las variables, en los modelos con y sin la variable de confusión
- Variables con **relevancia biológica**

Estrategia para la construcción del modelo

Paso 4: Comprobar el **supuesto de linealidad en el logit** para las variables continuas

Variables que no cumplen este supuesto:

- **Transformación** de variables continuas
- **Categorización** de variables continuas

Paso 5: Evaluar las **interacciones** entre las variables del modelo

- Seleccionar previamente las interacciones que puedan tener un **sentido clínico**

Ejemplo: Construcción de un modelo. Paso 1

```
> ## Fichero Datos: Bajo peso al nacer
> xx <- read.csv(file="C://Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
>
> ## Indice de las variables predictoras
>
> ind.var.pred <- c (3, 4, 5, 6, 8, 9, 10, 11)
>
> for ( ind in ind.var.pred )
+ {
+   out.1 <- glm( bajo_pes ~ xx[ , ind] , data=xx, family = binomial )
+   anova.1 <- anova(out.1 , test="Chisq")
+   print ( paste ( names(xx)[ind] ,
+                 round(anova.1$P[2], dig=5) , sep=" " ))
+ }
[1] "edad 0.09665"
[1] "peso 0.01446"
[1] "raza 0.08166"
[1] "fumador 0.02737"
[1] "hta 0.04491"
[1] "irr_urin 0.02426"
[1] "visi_med 0.37925"
[1] "part_pre2 0.00035"
```

- **Paso 1: análisis univariante**
 - Solo la variable “Número de visitas al medico” tiene una P>0.25, y es descartada en este paso

Ejemplo: Construcción de un modelo. Paso 2

```
> out.2 <- glm( bajo_pes ~ edad + peso + raza + fumador + hta + irr_urin  
+                               + part_pre2 , data=xx, family = binomial )  
> summary(out.2)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.63691	1.23028	0.518	0.60467
edad	-0.03775	0.03781	-0.998	0.31808
peso	-0.03287	0.01552	-2.118	0.03419 *
raza2	1.21274	0.53248	2.278	0.02275 *
raza3	0.80412	0.44843	1.793	0.07294 .
fumador1	0.84640	0.40806	2.074	0.03806 *
hta1	1.83869	0.70324	2.615	0.00893 **
irr_urin1	0.71113	0.46311	1.536	0.12465
part_pre2	1.22175	0.46301	2.639	0.00832 **

>

- **Paso 2: modelo multivariante completo**

- Hay 2 variables que no son estadísticamente significativas en este modelo:
“edad” ($P=0.318$), e “irritabilidad urinaria” ($P=0.125$) (**Test de Wald**)
- Primero se quita “edad” porque el P-valor es mayor

Ejemplo: Construcción de un modelo. Paso 2

```
> ## LRT Test del cociente de verosimilitudes
> drop1(out.2, test="Chisq")
Single term deletions

Model:
bajo_pes ~ edad + peso + raza + fumador + hta + irr_urin + part_pre2
      Df Deviance   AIC    LRT Pr(Chi)
<none>     196.83 214.83
edad       1  197.85 213.85 1.0179 0.313026
peso        1  201.83 217.83 4.9996 0.025353 *
raza        2  203.24 217.24 6.4066 0.040628 *
fumador     1  201.25 217.25 4.4133 0.035659 *
hta         1  204.01 220.01 7.1793 0.007375 **
irr_urin    1  199.15 215.15 2.3177 0.127909
part_pre2   1  203.95 219.95 7.1144 0.007647 **
>
```

- **Paso 2: modelo multivariante completo**

- Con el **test del cociente de verosimilitudes** se obtienen los mismos resultados: “edad” ($P=0.313$), e “irritabilidad urinaria” ($P=0.128$)
- Primero se quita “edad” porque el P-valor es mayor

Ejemplo: Construcción de un modelo. Paso 3

```
> out.3 <- glm( bajo_pes ~ peso + raza + fumador + hta + irr_urin  
+                               + part_pre2 , data=xx, family = binomial )  
> summary(out.3)  
  
Coefficients:  
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.12533   0.96756 -0.130  0.89694  
peso         -0.03509   0.01533 -2.289  0.02207 *  
raza2        1.30086   0.52848  2.461  0.01384 *  
raza3        0.85441   0.44091  1.938  0.05264 .  
fumador1     0.86658   0.40447  2.143  0.03215 *  
hta1         1.86690   0.70737  2.639  0.00831 **  
irr_urin1    0.75065   0.45882  1.636  0.10183  
part_pre2    1.12886   0.45039  2.506  0.01220 *  
---
```

- **Paso 3: modelo multivariante preliminar de efectos principales**
 - Quitando la edad, los coeficientes de regresión cambian poco
 - Edad es importante desde el punto de **vista biológico**
 - Se decide dejarla en el modelo, pendiente de analizar su escala y si participa en alguna interacción

Ejemplo: Construcción de un modelo. Paso 3

```
> out.4 <- glm( bajo_pes ~ edad + peso + raza + fumador + hta  
+                               + part_pre2 , data=xx, family = binomial )  
> summary(out.4)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.92491	1.20269	0.769	0.44187
edad	-0.04278	0.03757	-1.139	0.25475
peso	-0.03403	0.01553	-2.191	0.02843 *
raza2	1.16845	0.53258	2.194	0.02824 *
raza3	0.81462	0.44274	1.840	0.06578 .
fumador1	0.85833	0.40479	2.120	0.03397 *
hta1	1.74051	0.70310	2.475	0.01331 *
part_pre2	1.33397	0.45757	2.915	0.00355 **

- **Paso 3: modelo multivariante preliminar de efectos principales**

- Quitando “irritabilidad urinaria”, los cambios en los coeficientes de regresión no son muy grandes, y no podemos concluir que sea una variable de confusión
- En este ejemplo decidimos dejar la variable: se considera su **importancia biológica**, y que el OR estimado es grande (OR=2, P=0.125), aunque no sea estadísticamente significativo

Ejemplo: Construcción de un modelo. Paso 4

```
> ## Categorización de Peso
> quantile ( xx$peso )
    0%      25%      50%      75%     100%
36.28776 49.89567 54.88524 63.50358 113.39926
> xx$peso4gr <- cut ( xx$peso, breaks = c(quantile(xx$peso)[1:4], 999) ,
+                         labels=1:4 , right=F)
> table(xx$peso4gr)
 1  2  3  4
42 50 47 50
> ## Categorización de Edad
> quantile ( xx$edad )
    0%   25%   50%   75% 100%
 14    19    23    26   45
> xx$edad4gr <- cut ( xx$edad, breaks = c(quantile(xx$edad)[1:4], 999) ,
+                         labels=1:4 , right=F)
> table(xx$edad4gr)
 1  2  3  4
35 59 41 54
```

- **Categorización de variables continuas**
 - La función ***quantile()*** calcula los cuantiles
 - La función ***cut()*** categoriza según los puntos de corte (breaks) indicados. El parámetro *right=F* para intervalos cerrados a la izda y abiertos a la dcha [)

Ejemplo: Construcción de un modelo. Paso 4

```
> out.5 <- glm( bajo_pes ~ edad + peso4gr + raza + fumador + hta + irr_urin  
+                               + part_pre2 , data=xx, family = binomial )  
> summary(out.5)  
Coefficients:  
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.40188   1.00100 -0.401  0.68807  
edad         -0.04475   0.03808 -1.175  0.23997  
peso4gr2     -0.81400   0.48300 -1.685  0.09193 .  
peso4gr3     -0.71063   0.50311 -1.412  0.15782  
peso4gr4     -1.05346   0.54004 -1.951  0.05109 .  
raza2        1.10525   0.52031  2.124  0.03365 *  
raza3        0.79334   0.45020  1.762  0.07803 .  
fumador1    0.79764   0.40659  1.962  0.04979 *  
hta1         1.52905   0.67014  2.282  0.02251 *  
irr_urin1   0.65798   0.46624  1.411  0.15817  
part_pre2   1.28384   0.46389  2.768  0.00565 **
```

- **Paso 4: Variables continua. Supuesto de linealidad. Escala**
 - Los coeficientes de regresión de los cuartiles de “**peso**” son: - 0.81, - 0.71 y -1.05 todas las categorías son protectoras con respecto a la 1^a en un orden de magnitud parecido. No hay relación lineal
 - Se transforma como una **variable binaria**, donde la categoría de riesgo son las mujeres con peso < 49.5Kg. que codificaremos con 1

Ejemplo: Construcción de un modelo. Paso 4

```
> out.6 <- glm( bajo_pes ~ edad4gr + peso + raza + fumador + hta + irr_urin  
+                               + part_pre2 , data=xx, family = binomial )  
> summary(out.6)  
Coefficients:  
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.27678   0.99634 -0.278   0.7812  
edad4gr2     -0.17834   0.51665 -0.345   0.7300  
edad4gr3      0.29901   0.54258  0.551   0.5816  
edad4gr4     -0.21557   0.55779 -0.386   0.6991  
peso          -0.03197   0.01585 -2.017   0.0437 *  
raza2         1.23339   0.53806  2.292   0.0219 *  
raza3         0.83797   0.44223  1.895   0.0581 .  
fumador1      0.93156   0.41009  2.272   0.0231 *  
hta1          1.85321   0.71252  2.601   0.0093 **  
irr_urin1     0.80784   0.47139  1.714   0.0866 .  
part_pre2     1.09928   0.46207  2.379   0.0174 *
```

- **Paso 4: Variables continua. Supuesto de linealidad. Escala**
 - Los coeficientes de regresión de los cuartiles de “edad” son: - 0.18, 0.30 y - 0.22. No hay relación lineal, ni sugiere una transformación con puntos de corte, pero hay que tener en cuenta que no podemos considerar los coeficientes distintos de 0
 - Se mantiene edad como **variable continua**

Ejemplo: Construcción de un modelo. Paso 4

```
> xx$peso_gr <- NA  
> xx$peso_gr [ xx$peso < 49.5 ] <- 1  
> xx$peso_gr [ xx$peso > 49.5 ] <- 0  
> out.7 <- glm( bajo_pes ~ edad + peso_gr + raza + fumador + hta + irr_urin  
+                         + part_pre2 , data=xx, family = binomial )  
> summary(out.7)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.21678	0.95567	-1.273	0.20294
edad	-0.04648	0.03739	-1.243	0.21381
peso_gr	0.84206	0.40553	2.076	0.03785 *
raza2	1.07346	0.51507	2.084	0.03715 *
raza3	0.81537	0.44529	1.831	0.06709 .
fumador1	0.80720	0.40444	1.996	0.04595 *
hta1	1.43523	0.64827	2.214	0.02683 *
irr_urin1	0.65763	0.46662	1.409	0.15873
part_pre2	1.28168	0.46211	2.774	0.00555 **

- **Paso 4: Modelo de efectos principales**
 - Edad como continua
 - Peso como binaria

Ejemplo: Construcción de un modelo. Paso 5

```
> ## Explorando las interacciones con peso_gr
> set.var <- c( "edad", "raza", "fumador", "hta" ) ## Variables explorar interacción
>
> # Bucle para cada variable a explorar
> for ( name.var in set.var )
+ {
+   ## Se construye un texto con el modelo a evaluar
+   modelo <- paste( "bajo_pes ~ edad + raza + fumador + hta+ irr_urint part_pre2 + ",
+                     name.var , "* peso_gr" )
+   ## Modelo con interacción
+   out.int <- glm ( as.formula(modelo), data=xx, family = binomial )
+   ## LRT del modelo de efectos principales y el que incluye la interacción
+   p.value.inter <- anova ( out.7, out.int, test="Chisq")$"Pr(>Chi)"[2]
+   print( paste( name.var , " * peso_gr ", round(p.value.inter, dig=4), sep=""))
+ }
[1] "edad * peso_gr 0.1101"
[1] "raza * peso_gr 0.1831"
[1] "fumador * peso_gr 0.1266"
[1] "hta * peso_gr 0.2908"
```

- Se decide que las interacciones que pueden tener sentido biológico son las del peso de la madre con otras variables importantes: edad, raza, fumador y hta; y se decide explorarlas todas
- Hay dos interacciones con P-valor cerca de 0.1 que se deciden explorar en el modelo multivariante

Ejemplo: Construcción de un modelo. Paso 5

```
> out.8 <- glm( bajo_pes ~ edad + peso_gr + raza + fumador + hta + irr_urin + part_pre2  
+                           edad*peso_gr + peso_gr*fumador , data=xx, family = binomial )  
> summary(out.8)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.51175	1.08754	-0.471	0.63795
edad	-0.08398	0.04557	-1.843	0.06533 .
peso_gr	-1.72995	1.86831	-0.926	0.35447
raza2	1.08310	0.51892	2.087	0.03687 *
raza3	0.75968	0.46403	1.637	0.10161
fumador1	1.15313	0.45844	2.515	0.01189 *
hta1	1.35922	0.66147	2.055	0.03989 *
irr_urin1	0.72817	0.47948	1.519	0.12885
part_pre2	1.23158	0.47139	2.613	0.00898 **
edad:peso_gr	0.14741	0.08286	1.779	0.07523 .
peso_gr:fumador1	-1.40738	0.81868	-1.719	0.08560 .

- **Paso 5: Interacciones**
 - Se han incluido las 2 interacciones con sentido biológico y con significación estadística al 0.1

Ejemplo: Análisis de influencia. Distancia de Cook

```
> ## Análisis de influencia. Distancia de Cook
> cook.8 <- cooks.distance(out.8)
> max(cook.8)
[1] 0.07854959
```

- La función ***cooks.distance()*** se puede ejecutar con modelos de regresión lineal *lm()* y con modelos de regresión logística *glm()*
- No hay ninguna observación influyente en el ajuste del modelo de regresión logística
- El $\max(D_i)$ es 0.08 muy lejos de 1

Ejercicios

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
1. Ajustar e interpretar el modelo de **regresión logística multivariante completo**
 - AGE + BECK + IVHX + NDRUGTX + RACE + TREAT + SITE
 2. Ajustar e interpretar el modelo de **regresión logística multivariante de efectos principales**, quitando las variables que no son significativas, pero dejando SITE y RAZA por su importancia biológica
 3. Analizar la escala de la **variable continua NDRUGTX** en ese modelo de efectos principales, definiendo una variable con los quartiles

Estadística Aplicada a la Investigación Biomédica con R

17 Análisis de Supervivencia

- ✓ Introducción al análisis de supervivencia
- ✓ Estimador de Kaplan-Meier
- ✓ Análisis descriptivo del tiempo de supervivencia
- ✓ Comparación de curvas de supervivencia

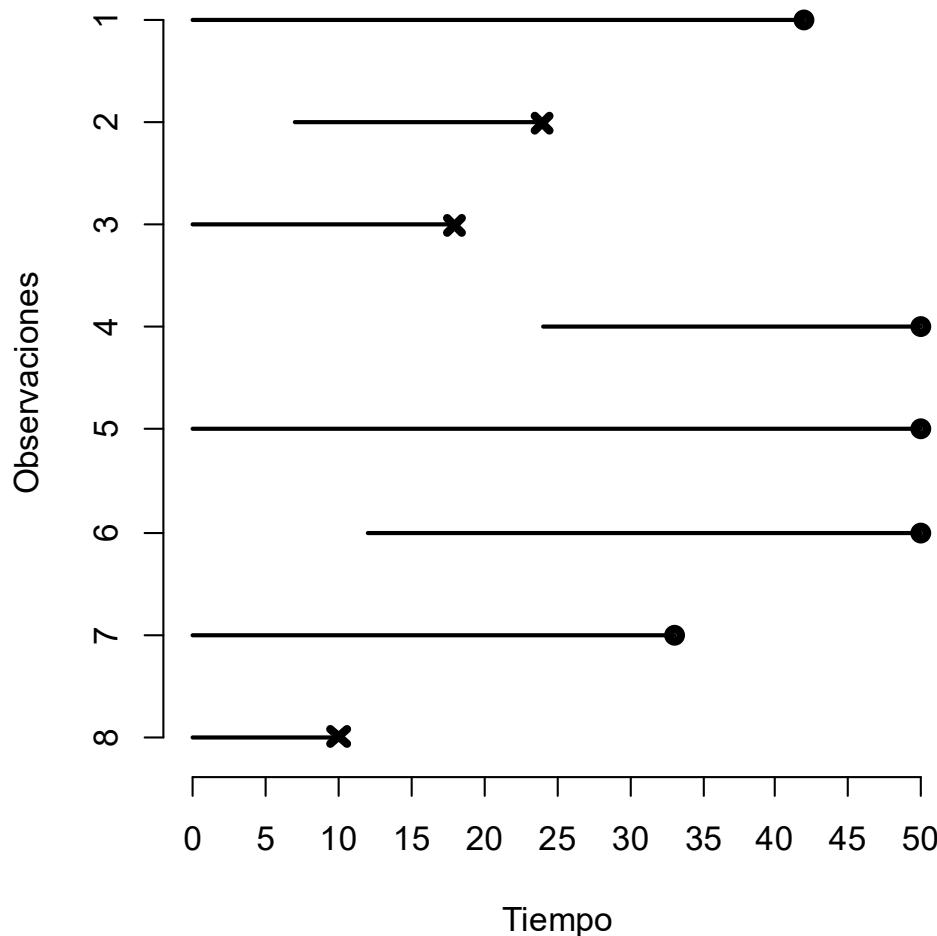
Análisis de supervivencia

- El **análisis de supervivencia** reúne las técnicas estadísticas apropiadas para analizar estudios en los que los individuos son seguidos **a lo largo de un periodo de tiempo** hasta que ocurre un determinado **evento de interés**
 - Ejemplos de **eventos clínicos**: recidiva, recaída, progresión, muerte, alta hospitalaria, curación,
- La **variable respuesta** a analizar es el **tiempo hasta que ocurre el evento**
- El objetivo del Análisis de Supervivencia es describir **la probabilidad de ocurrencia del evento** a lo largo del tiempo
 - Evolución de la tasa de incidencia del evento, tasa de riesgo
- **Estudios de seguimiento**
 - Fechas de inicio y final del estudio
 - Periodo de reclutamiento, en el que los individuos se incorporan al estudio

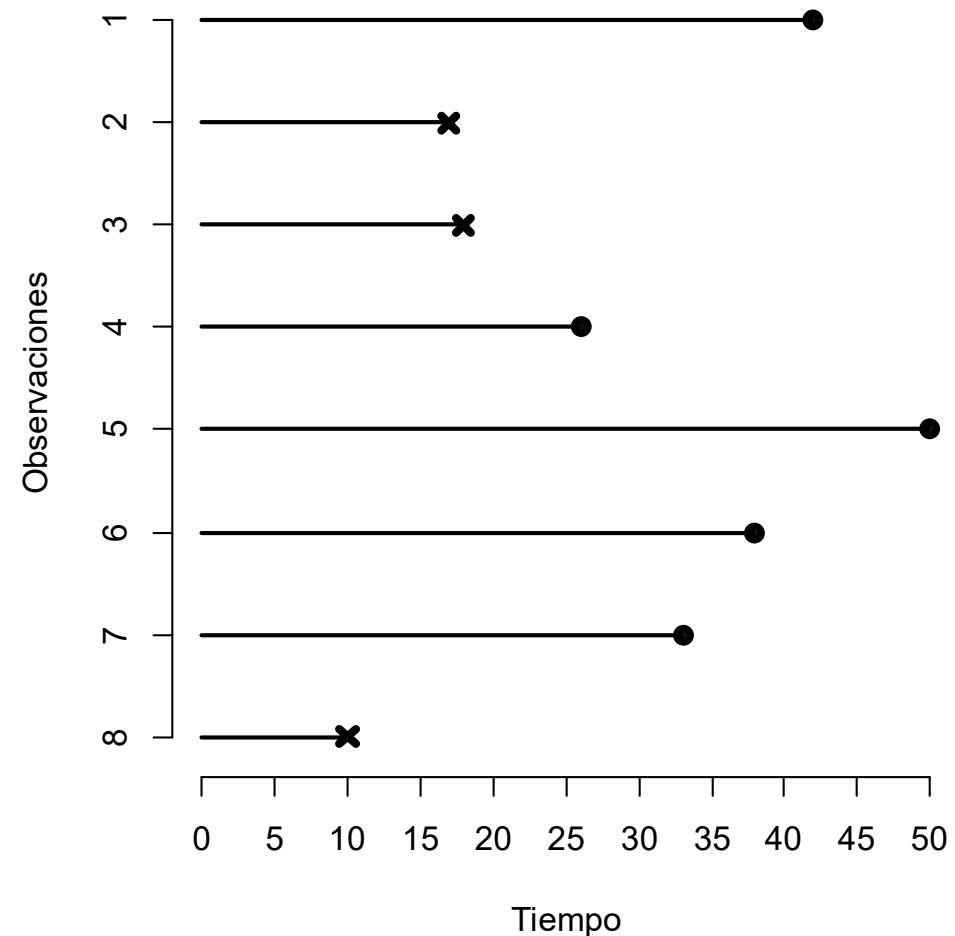
Tiempo de supervivencia

- **Tiempo de seguimiento**
 - Tiempo transcurrido entre la fecha de **incorporación** al estudio y la fecha de la **última observación**
- **Observaciones censuradas**
 - Individuos para los que **no ha ocurrido el evento**
 - Información que se puede incorporar al análisis
 - **Censuras a la derecha:** individuos en los que no ha ocurrido el evento antes de finalizar el estudio, o individuos perdidos en el seguimiento por otras causas
- **Tiempo de supervivencia**
 - Tiempo entre la **incorporación** al estudio y la fecha en la que ha **ocurrido el evento**, en los casos en los que ha ocurrido
 - Se llaman **tiempos completos** o no censurados

Tiempo de supervivencia



x – evento
o – censurado



tiempo de supervivencia
tiempo de seguimiento

Análisis de supervivencia

- La **variable respuesta** en un análisis de supervivencia tiene dos componentes:
 - c_i es una variable **binaria**, llamada también **estado (status)**
 - 1 si ha ocurrido el evento
 - 0 si es una observación censurada
 - t_i es el **tiempo de seguimiento**, que coincide con **el tiempo de supervivencia** para las observaciones para las que ha ocurrido el evento
- El análisis de supervivencia incluye el análisis del “ritmo” o “velocidad” en la que se presenta el evento en el tiempo, es decir, la **tasa de incidencia del evento**
 - El análisis con regresión logística del evento como variable binaria pierde esta información del “ritmo”
 - La variable tiempo no se puede analizar con regresión lineal, porque existen observaciones censuradas

Función de supervivencia

- T variable aleatoria **tiempo de supervivencia** (cuantitativa positiva, $T>0$)
- Función de distribución:

$$F(t) = \text{Prob}(T \leq t)$$

- **Función de supervivencia:** probabilidad de que un individuo sobreviva durante un tiempo superior a t
 - Al inicio del estudio **$S(0)=1$** porque todos los individuos están vivos
 - **$S(t)$ disminuye**, y podría llegar a 0 dependiendo de las observaciones censuradas, o si se ha producido el evento en todas las observaciones

$$S(t) = \text{Prob}(T > t) = 1 - F(t)$$

- El primer objetivo de un análisis de supervivencia es una **descripción univariante** de los datos observados mediante la **estimación de la función de supervivencia**

Estimador no paramétrico. Kaplan-Meier

- El **método de Kaplan-Meier**
 - estima las probabilidades de supervivencia $S(t_j)$ **para cada tiempo completo**, en los instantes en los que ha ocurrido el evento
 - para sobrevivir en un momento determinado, se ha tenido que haber **sobrevivido en todos los tiempos anteriores** hasta ese momento
- Se basa en una **probabilidad condicional** compuesta
 - Para cada instante de tiempo, la supervivencia se calcula como producto de la supervivencia en el instante anterior y la tasa de supervivencia en ese instante

$$\hat{S}(t_j) = \hat{S}(t_{j-1}) \cdot \hat{S}(t_j / t_{j-1})$$

Estimador no paramétrico. Kaplan-Meier

- **Estimador de Kaplan-Meier**
 - n observaciones, donde en m observaciones ha ocurrido el evento ($n \geq m$)
 - $t_{(1)} \leq t_{(2)} \leq \dots \leq t_{(m)}$ tiempos de supervivencia ordenados (tiempos completos)
 - n_i número de individuos en riesgo en el tiempo $t_{(i)}$; d_i número observado de eventos
- El **estimador de Kaplan-Meier** de la función de supervivencia en el tiempo t

$$\hat{S}(t) = \prod_{t_{(i)} \leq t} \frac{n_i - d_i}{n_i}$$

- en cada tiempo t se obtiene **multiplicando** una secuencia de estimaciones de **probabilidades condicionales** de supervivencia, para los **tiempos anteriores a t**
- cada probabilidad condicional se obtiene **dividiendo** el número de individuos que estaban **en riesgo al final** del intervalo con los que lo estaban **al principio**
- las **observaciones censuradas** influyen en el cálculo de los individuos en riesgo

Medidas descriptivas del tiempo de supervivencia

- **Media** del tiempo de supervivencia
 - no se suele calcular porque la variable tiempo no es simétrica
 - no se dispone del tiempo para todos los individuos (observaciones censuradas)
- **Mediana** del tiempo de supervivencia
 - no se necesita conocer el tiempo de supervivencia de todos los individuos

$$\hat{t}_{50} = \min \left\{ t / \hat{S}(t) \leq 0.50 \right\} \quad S(t_{\text{mediana}}) = 0.50$$

- **Cuartiles** y percentiles son medidas también adecuadas
 - se podrán calcular dependiendo del mínimo valor del estimador KM
 - tiempo en el que sobrevive una proporción de la población

$$\hat{t}_p = \min \left\{ t / \hat{S}(t) \leq (p/100) \right\}$$

Ejemplo: Análisis descriptivo tiempo supervivencia

```

> library(survival)
> ## Fichero Datos: whas500
> xx <- read.csv(file="C://Bioestadistica con R/Datos/whas500.csv", sep=";")
> ## Función que "empaquetá" los tiempos y eventos, y se usa para los análisis
> xx.surv <- Surv(xx$lenfol,xx$fstat)
>
> ## KM estimator
> surv.all <- survfit ( Surv(lenfol, fstat) ~ 1 , data=xx )
> summary( surv.all )
Call: survfit(formula = xx.surv ~ 1, data = xx)

time n.risk n.event survival std.err lower 95% CI upper 95% CI
 1    500      8   0.984 0.00561      0.9731     0.995
 2    492      8   0.968 0.00787      0.9527     0.984
 3    484      3   0.962 0.00855      0.9454     0.979
 4    481      2   0.958 0.00897      0.9406     0.976
 5    479      2   0.954 0.00937      0.9358     0.973
 6    477      5   0.944 0.01028      0.9241     0.964
 7    472      6   0.932 0.01126      0.9102     0.954
10    466      3   0.926 0.01171      0.9033     0.949
11    463      4   0.918 0.01227      0.8943     0.942
          . . . . . . . . . . .
          . . . . . . . . . . .

> surv.all    ## mediana

records  n.max n.start  events  median 0.95LCL 0.95UCL
      500      500      500     215     1627     1527       NA

```

Ejemplo: Análisis descriptivo tiempo supervivencia

```
> summary(surv.all)
```

time	n.risk	n.event	survival	std.err	lower	95% CI	upper	95% CI
1	500	8	0.984	0.00561	0.9731	0.995		
2	492	8	0.968	0.00787	0.9527	0.984		
3	484	3	0.962	0.00855	0.9454	0.979		
	
	
295	376	1	0.750	0.01936	0.7130	0.789		
297	375	2	0.746	0.01947	0.7088	0.785		
312	373	1	0.744	0.01952	0.7067	0.783		
	
	
1624	91	1	0.505	0.02798	0.4531	0.563		
1627	90	1	0.499	0.02822	0.4471	0.558		
1671	89	1	0.494	0.02846	0.4411	0.553		
	
	
2350	3	1	0.292	0.12325	0.1277	0.668		
2353	2	1	0.146	0.12025	0.0291	0.733		
2358	1	1	0.000	NaN	NA	NA		

- Mediana = 1627 y los cuartiles Q3 (P_{75}) = 297 y Q1 (P_{25}) = 2353

Ejemplo: Estimador de Kaplan-Meier

```
> summary( surv.all )
Call: survfit(formula = xx.surv ~ 1, data = xx)

  time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1    500      8     0.984 0.00561      0.9731      0.995
    2    492      8     0.968 0.00787      0.9527      0.984
    3    484      3     0.962 0.00855      0.9454      0.979
    4    481      2     0.958 0.00897      0.9406      0.976
    5    479      2     0.954 0.00937      0.9358      0.973
    6    477      5     0.944 0.01028      0.9241      0.964
    . . . . . . . . . . .
    . . . . . . . . . . .
```

- El estimador de Kaplan-Meier se puede calcular manualmente:

$$0.984 = \frac{500 - 8}{500}$$

$$\hat{S}(t_j) = \hat{S}(t_{j-1}) \cdot \hat{S}(t_j / t_{j-1})$$

$$0.968 = 0.984 \cdot \frac{492 - 8}{492}$$

$$\hat{S}(t) = \prod_{t_{(i)} \leq t} \frac{n_i - d_i}{n_i}$$

$$0.962 = 0.968 \cdot \frac{484 - 3}{484}$$

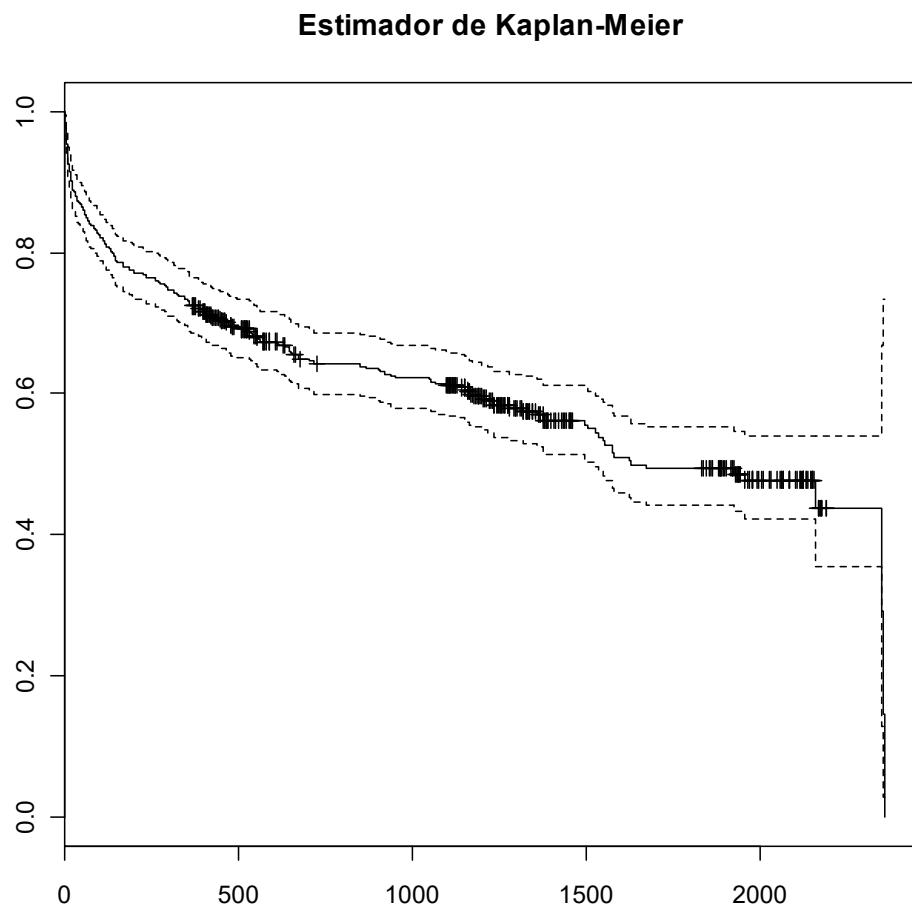
Ejemplo: Estimador de Kaplan-Meier

```
> ## Supervivencia cada año. Tablas de vida
> summary( surv.all, time=seq(0,3000,365))
   time n.risk n.event survival std.err lower 95% CI upper 95% CI
      0     500       0    1.000  0.0000    1.000    1.000
  365     362     138    0.724  0.0200    0.686    0.764
    730     236      34    0.642  0.0222    0.599    0.687
   1095     226      10    0.614  0.0229    0.571    0.661
   1460     100      15    0.561  0.0248    0.515    0.612
   1825      88      12    0.494  0.0285    0.441    0.553
   2190       5       3    0.438  0.0468    0.355    0.540

> ## Curva de Supervivencia
> dev.new()
> plot(surv.all, main="Estimador de Kaplan-Meier")
```

- Con **summary()** se pueden obtener estimaciones con el **método de la tabla de vida** que es similar a KM, pero con tiempos agrupados que se especifican en el parámetro *time*=
- Con la función **plot()** se muestra la curva de supervivencia obtenida con el estimador Kaplan-Meier

Ejemplo: Estimador de Kaplan-Meier



- La **curva KM** estima la función de supervivencia, probabilidad acumulada de que no haya ocurrido el evento
- Es una curva **decreciente escalonada** (donde ha ocurrido algún evento)
- Empieza en 1, y acaba en 0 si el último tiempo observado corresponde a un individuo en el que ha ocurrido el evento
- La **mediana**, 1627 es el tiempo en el que $S(1627)=0.50$
- Las cruces representan a las observaciones censuradas

Comparación de curvas de supervivencia

- **Analizar si la supervivencia** en 2 o más grupos es igual o si hay diferencias estadísticamente significativas entre los grupos
- Representar las **gráficas KM** de los grupos
- Se compara el **número de eventos observados** en cada uno de los k grupos con el **número de eventos esperados**, en cada tiempo de supervivencia, tiempos en los que ha ocurrido algún evento
 - El número de eventos esperados se calculará suponiendo que la supervivencia es igual en todos los grupos

Comparación de curvas de supervivencia

- Para cada uno de los **tiempos de supervivencia observados**, se puede definir la **tabla de contingencia** entre la variable que define los grupos y la variable estado:

Tabla para el tiempo $t_{(i)}$

Estado	Grupo		
	$X = 1$	$X = 0$	Total
$E = 1$	d_{1i}	d_{0i}	d_i
$E = 0$	$n_{1i} - d_{1i}$	$n_{0i} - d_{0i}$	$n_i - d_i$
En riesgo	n_{1i}	n_{0i}	n_i

- El número de **individuos en riesgo** es denotado como n_{0i} y n_{1i} para los grupos 0 y 1
- El número de **eventos observados** es denotado como d_{0i} y d_{1i} para los grupos 0 y 1

- El número de **eventos esperados** en cada tiempo será

$$\hat{e}_{1i} = \frac{n_{1i}d_i}{n_i}$$

d_i/n_i es la proporción de eventos que han ocurrido en $t_{(i)}$
Para calcular el número de eventos que le corresponderían al grupo 1, hay que multiplicar por su tamaño muestral n_{1i}

Prueba log-rank (Mantel-Haenszel)

- El **test log-rank**, es la suma de las diferencias entre los eventos observados y los eventos esperados para todos los tiempos de supervivencia observados, dividido por una estimación de la varianza

$$Q = \frac{\left[\sum_{i=1}^m (d_{li} - \hat{e}_{li}) \right]^2}{\sum_{i=1}^m \hat{v}_{li}}$$

- Bajo la hipótesis nula (las 2 curvas de supervivencia son iguales), Q sigue una distribución chi-cuadrado con 1 gl
- Generalización a **k grupos** ($k > 2$), Q sigue una chi-cuadrado con $k-1$ gl
 - El estadístico del test es más complicado porque contiene las varianzas y covarianzas entre todos los grupos

Comparación de curvas de supervivencia

- **Familia de tests**, incluyendo un peso w_i que pondera los términos del sumatorio

$$Q = \frac{\left[\sum_{i=1}^m w_i (d_{li} - \hat{e}_{li}) \right]^2}{\sum_{i=1}^m w_i^2 \hat{V}_{li}}$$

- **Test de Wilcoxon** usa como pesos el número de individuos en riesgo ($w_i=n_i$)
 - Da más importancia a los **tiempos iniciales** donde hay más individuos
- **Variables cuantitativas**, se suelen **categorizar** en cuartiles o terciles
 - Se pueden mostrar los **estimadores KM**, y realizar la **prueba log-rank**

Ejemplo: Comparación de curvas

```
> surv.sexo <- survfit ( Surv(lenfol/365, fstat) ~ gender, data=xx )
> surv.sexo ##median
      records n.max n.start events median 0.95LCL 0.95UCL
gender=0     300    300     300    111    5.92     4.58      NA
gender=1     200    200     200    104    3.61     2.48     4.46
> summary(surv.sexo)
. . .
> ## Test log-rank
> logrank.test <- survdiff ( Surv(lenfol/365,fstat) ~ gender, data=xx)
> logrank.test

      N Observed Expected (O-E)^2/E (O-E)^2/V
gender=0 300       111     130.7      2.98      7.79
gender=1 200       104      84.3      4.62      7.79
  Chisq= 7.8 on 1 degrees of freedom, p= 0.00525
> ## Test Peto-Peto (modificación del de Wilcoxon)
> survdiff ( Surv(lenfol/365,fstat) ~ gender, data=xx, rho=1)

      N Observed Expected (O-E)^2/E (O-E)^2/V
gender=0 300      87.2     101.8      2.09      6.73
gender=1 200      79.3      64.7      3.28      6.73
  Chisq= 6.7 on 1 degrees of freedom, p= 0.00948
```

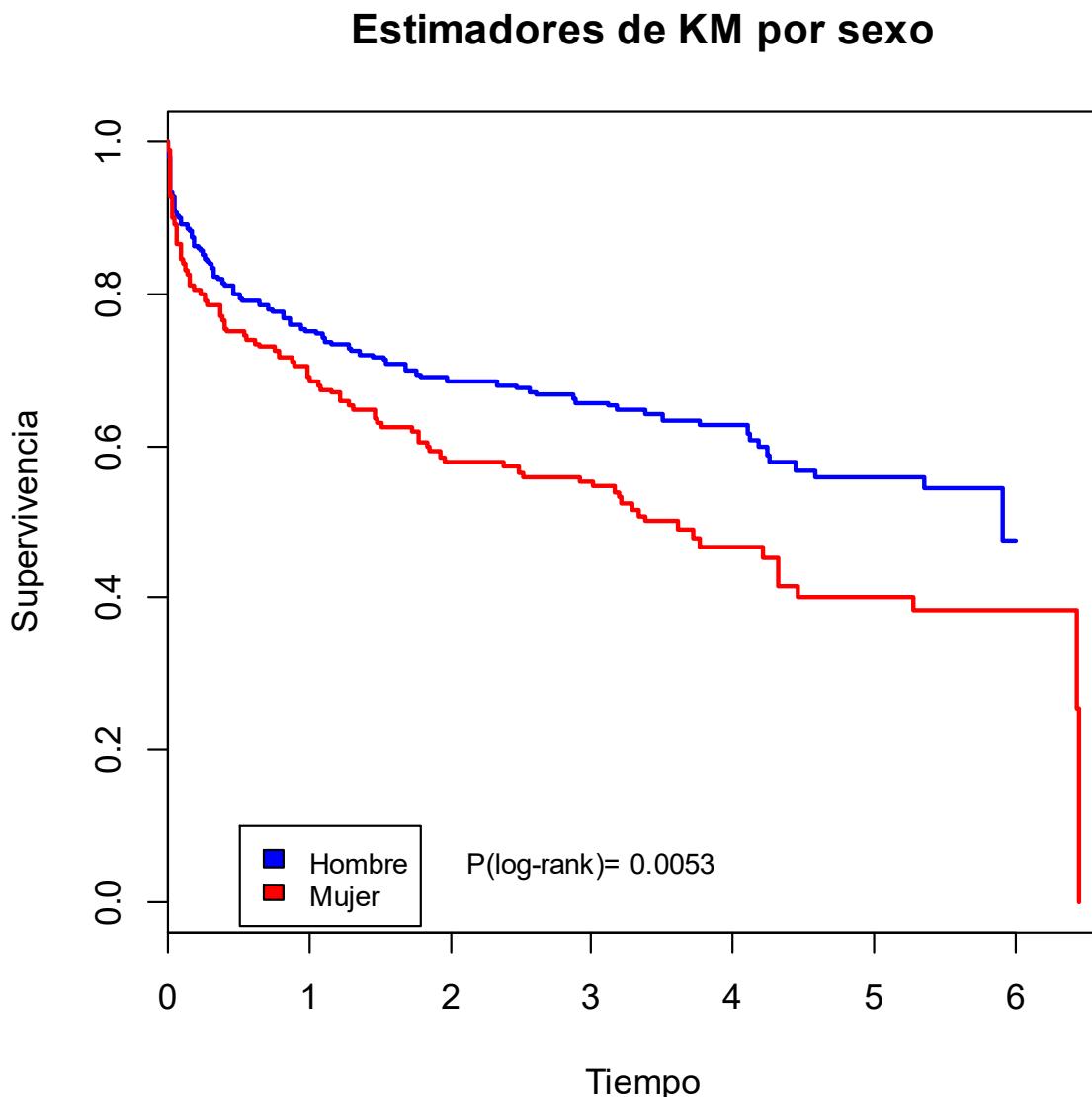
- Hay **diferencias significativas** en la supervivencia por sexo ($P=0.005$). Las mujeres presentan una peor supervivencia: la mediana de la supervivencia en las mujeres es 3.61 frente a 5.92 en los hombres

Ejemplo: Comparación de curvas

```
> ## Curvas KM por sexo
> dev.new()
> plot ( surv.sexo, main="Estimadores de KM por sexo",
+         xlab="Tiempo", ylab="Supervivencia",
+         conf.int=F, col=c("blue","red"), mark.time=F, lwd=2)
>
> legend(0.5,0.10, c("Hombre","Mujer"), c("blue","red"), cex = 0.8)
>
> text( 3,0.05, cex=0.8, paste("P(log-rank)=", 
+         round( 1-pchisq(logrank.test$chisq, length(logrank.test$obs)-1 ), dig=4) ) )
```

- La función ***plot()*** usa como argumento el resultado de ***survfit()*** donde se han obtenido estimaciones de la supervivencia para cada una de las categorías de la variable “sexo”

Ejemplo: Comparación de curvas



Ejercicios

- Fichero de datos: actg320.csv
 - Variable tiempo: “time”
 - Variable status: “censor”
1. Describe la **supervivencia global** para toda la muestra
 - Gráfico de KM
 - Mediana
 2. Compara la **supervivencia** para los pacientes que se les suministró el nuevo tratamiento (variable “tx”)
 - Test log-rank
 - Gráficos KM para los 2 grupos

Estadística Aplicada a la Investigación Biomédica con R

18-19 Regresión de Cox

- ✓ **Modelo de Cox de riesgos proporcionales**
- ✓ **Interpretación y estimación de los parámetros**
- ✓ **Significación de los coeficientes de regresión**
- ✓ **Variables predictoras categóricas**
- ✓ **Variables de confusión**
- ✓ **Interacciones**
- ✓ **Estrategia para la construcción de un modelo de Cox**
- ✓ **Modelo de Cox estratificado**

Función de riesgo

- La función de riesgo (“hazard function”) indica la tasa de mortalidad instantánea en t condicionada a que se haya sobrevivido hasta el instante anterior
 - “Fuerza de la mortalidad”. Medida de la tendencia a fallecer en un instante
 - Es la probabilidad condicionada por unidad de tiempo que tiene un individuo de fallecer en un instante t si había sobrevivido hasta el instante anterior

$$h(t) = \lim_{\delta \rightarrow 0} \frac{P(t \leq T \leq t + \delta / t \leq T)}{\delta} = - \frac{d \log S(t)}{dt}$$

- La función de riesgo es siempre **positiva** y no tiene límite.
- Es mayor cuando la curva de supervivencia presenta mayor caída
- La función de riesgo es útil para **modelar los datos de supervivencia**

Modelo de Cox de riesgos proporcionales

- El **modelo de regresión de Cox** permite:
 - **Modelar** la relación entre un conjunto de variables predictoras y **la tasa de riesgo** (hazard function)
 - **Predecir** la función de supervivencia para un individuo que presente unos determinados valores en las variables predictoras
- El **modelo de regresión de Cox** o **modelo de riesgos proporcionales** modela la **función de riesgo**: el riesgo de que se produzca el evento en el instante t para un individuo con unos valores X en las variables predictoras

$$h(t;X) = h_0(t) \cdot e^{\beta_1 X_1 + \dots + \beta_p X_p}$$

- el primer término caracteriza cómo la función de riesgo cambia en **el tiempo**
- el segundo término cómo cambia en función de los valores de los individuos en **las variables predictoras**

Modelo de Cox de riesgos proporcionales

- El modelo de regresión de Cox es **semiparamétrico** porque se estiman los parámetros β_j del modelo, pero la forma de $h_0(t)$ **no se especifica**

$$h(t; X) = h_0(t) \cdot e^{\beta_1 X_1 + \dots + \beta_p X_p}$$

- $h_0(t)$ se llama **función de riesgo de referencia (función baseline)**
 - Tasa instantánea de riesgo de un individuo con todos sus valores $X=0$
 - Si se toman X centradas, $h_0(t)$ sería el riesgo de un individuo “medio”, y el segundo término del modelo **modifica** ese riesgo “medio” para un patrón X
- Los parámetros del modelo de riesgos proporcionales β_1, \dots, β_p son estimados por el **método de verosimilitud parcial**
 - La función de verosimilitud está basada solo en las observaciones donde ha ocurrido el evento; pero las observaciones censuradas intervienen en los cálculos de los individuos en riesgo en cada tiempo

Interpretación de los parámetros del modelo

- Si consideramos una variable predictora **binaria** X con códigos 0 y 1, tendríamos que el modelo de regresión de Cox sería:

$$h(t;X) = h_0(t) \cdot e^{\beta X}$$

- **exp(β)** es el **cociente entre las tasas instantáneas** de riesgo entre 2 individuos con valores X=1 y X=0

$$\frac{h(t;X=1)}{h(t;X=0)} = \frac{h_0(t) \cdot e^{\beta}}{h_0(t) \cdot e^0} = e^{\beta}$$

- Si la variable predictora X es **continua**, **exp(β)** sería el **cociente entre las tasas instantáneas** de dos individuos que difieren en 1 unidad en X
 - Comparando un individuo con valor x_1 en X con otro con valor $x_2 = x_1 + 1$

Interpretación de los parámetros del modelo

- De forma general, si se comparan 2 individuos con valores A y B en las variables predictoras ($x_{A1}, x_{A2}, \dots, x_{Ap}$) y ($x_{B1}, x_{B2}, \dots, x_{Bp}$) respectivamente

$$HR(t; X_A; X_B) = \frac{h(t; X = X_A)}{h(t; X = X_B)} = \frac{h_0(t) \cdot e^{(\beta_1 x_{A1} + \dots + \beta_p x_{Ap})}}{h_0(t) \cdot e^{(\beta_1 x_{B1} + \dots + \beta_p x_{Bp})}} = e^{\beta_1(x_{A1} - x_{B1})} \cdot \dots \cdot e^{\beta_p(x_{Ap} - x_{Bp})}$$

- El cociente se llama **razón de riesgos (hazard ratio, HR)** de los individuos con valores X_A en las variables predictoras respecto de los individuos con valores X_B
- La **razón de riesgos** solo depende de los **coeficientes** del modelo
- La razón de riesgos es **independiente del tiempo**, es decir, la contribución de las variables predictoras en la tasa instantánea de riesgo es la misma en cualquier momento del tiempo

Supuesto de riesgos proporcionales

- **Principio de riesgos proporcionales**
 - La **función de riesgo** para los individuos con un patrón X_A es **proporcional** a la función de riesgo para otro individuo con patrón X_B , **durante todo el tiempo de seguimiento**
- Hay **contrastos y gráficos** para validar si se cumple este supuesto, basados en estudiar si los coeficientes β_j son constantes en el tiempo
- Se analizan modelos que incluyen interacciones entre las variables predictoras y el tiempo o una función del tiempo, evaluando los coeficientes de esos términos

Test del cociente de verosimilitudes parciales

- **Significación global del modelo**

- Se evalúa el incremento en el logaritmo de la verosimilitud parcial, del modelo que se desea evaluar $LL(mod)$, con el modelo que solo contiene la constante, $L(0)$
- Bajo la hipótesis nula, G sigue una chi-cuadrado con p gl siendo p el nº variables

$$G = -2 \cdot \{ LL(0) - LL(mod) \} = -2 \cdot \ln PLR = -2 \cdot \ln \frac{L(0)}{L(mod)}$$

- **Significación de los coeficientes**

- Se evalúa el incremento en el logaritmo de la verosimilitud parcial, del modelo con p variables $LL(p)$, y el modelo quitando la variable que se desea evaluar $L(p-1)$
- Bajo la hipótesis nula ($H_0: \beta_j=0$), G sigue una chi-cuadrado con 1 gl

$$G = -2 \cdot \{ LL(p-1) - LL(p) \}$$

Test de Wald

- Se utiliza también para evaluar la significación de los coeficientes de regresión, y se basa en el cociente entre los estimadores de regresión y su error estándar (SE)
 - Bajo la hipótesis nula $H_0: \beta_j = 0$ el cociente sigue una distribución normal estándar

$$Z = \frac{\hat{\beta}_j}{\hat{SE}(\hat{\beta}_j)} \approx N(0,1)$$

- Este resultado se puede utilizar para definir **intervalos de confianza** para los estimadores de regresión

$$\hat{\beta}_j \pm z_\alpha \cdot \hat{SE}(\hat{\beta}_j)$$

$$\hat{\beta}_j \pm 1.96 \cdot \hat{SE}(\hat{\beta}_j) \quad \text{para } \alpha=0.05$$

Ejemplo: Regresión de Cox. Variable continua

```
> library(survival)
> ## Fichero Datos: whas500
> xx <- read.csv(file="C://Bioestadistica con R/Datos/whas500.csv", sep=";")
> ## Variable continua
> cox1 <- coxph ( Surv(lenfol, fstat) ~ age, data=xx )
> summary(cox1)
      coef exp(coef) se(coef)     z Pr(>|z|)
age 0.066339  1.068589 0.006079 10.91   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
age      1.069      0.9358    1.056      1.081

Rsquare= 0.247  (max possible= 0.993 )
Likelihood ratio test= 142.1  on 1 df,  p=0
Wald test            = 119.1  on 1 df,  p=0
Score (logrank) test = 126.6  on 1 df,  p=0
```

- El riesgo estimado de morir aumenta un 6.9% por cada año de aumento en la edad (HR=1.069, IC95%: 1.056 – 1.081 , P<0.001)
- La edad está asociada al riesgo de morir (IC95% no contiene el 1 y P < 0.05)
- Este aumento de riesgo es constante en todo el rango de la edad, es decir, es el mismo comparando dos pacientes de 30 y 31 años, o de 60 y 61
- Observación: en el modelo de Cox no hay término constante β_0

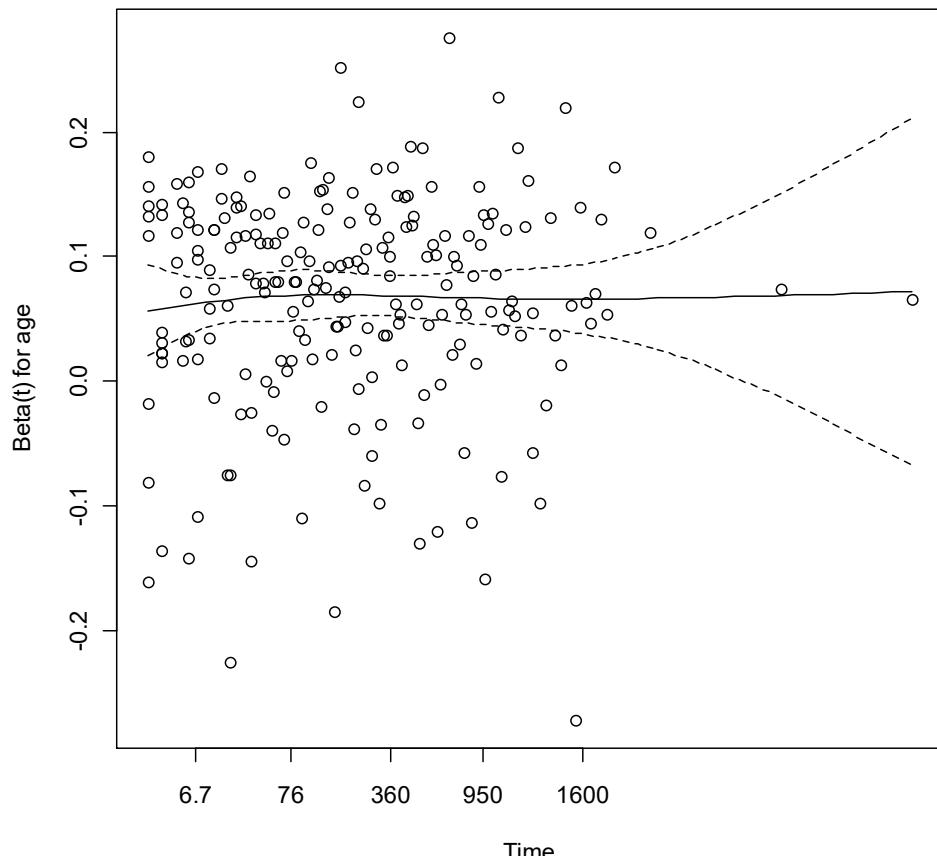
Ejemplo: Regresión de Cox. Variable continua

```
## Cambio de escala
> c <- 10
>
> summary(cox1)$coeff
      coef  exp(coef)    se(coef)      z Pr(>|z|)
age 0.06633895  1.068589  0.006079292 10.91228      0
> summary(cox1)$coeff[1]
[1] 0.06633895
> exp(summary(cox1)$coeff[1])
[1] 1.068589
> exp( c * summary(cox1)$coeff[1] )
[1] 1.941361
> exp( c * summary(cox1)$coeff[1] - 1.96 * abs(c) * summary(cox1)$coeff[3] )
[1] 1.723290
> exp( c * summary(cox1)$coeff[1] + 1.96 * abs(c) * summary(cox1)$coeff[3] )
[1] 2.187028
```

- El riesgo aumenta un 94% cuando comparamos 2 individuos con 10 años de diferencia (HR=1.94, IC95%: 1.72 – 2.19)

Ejemplo: Supuesto de riesgos proporcionales

```
## Supuesto de Riesgos Proporcionales
> cox.zph(cox1)
   rho chisq   p
age 0.0203 0.0865 0.769
> plot(cox.zph(cox1))
```



- No hay evidencia de que la variable edad no siga el supuesto de riesgos proporcionales ($P=0.769$)
- El gráfico muestra que el coeficiente beta es constante en el tiempo
- Se observa también que el IC del coeficiente no incluye el 0

Variables predictoras categóricas

- **Variable predictora binaria**
 - Usar 2 códigos consecutivos: 0-1 o 1-2
- **Variable predictora categóricas con k categorías ($k > 2$)**
 - Definir **k-1 variables dummy** (indicadoras) Z_1, \dots, Z_{k-1}
 - Categoría de referencia: la más numerosa o más protectora

Categorías	Z_1	Z_2	Z_3
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1

- **Variables categóricas ordinales**
 - Análisis de tendencia. Variable continua

Ejemplo: Regresión de Cox. Variable binaria

```
## Variable binaria
> cox2 <- coxph ( Surv(lenfol, fstat) ~ chf, data=xx )
> summary(cox2)

      coef exp(coef)  se(coef)      z Pr(>|z|)
chf 1.200     3.319    0.138 8.692   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
chf     3.319     0.3013    2.532     4.35

Concordance= 0.642  (se = 0.016 )
Rsquare= 0.135  (max possible= 0.993 )
Likelihood ratio test= 72.28 on 1 df,  p=0
Wald test            = 75.55 on 1 df,  p=0
Score (logrank) test = 84.6 on 1 df,  p=0
```

- El riesgo de morir es 3.3 veces mayor en los que presentaban complicaciones cardiacas (“chf”) con respecto a los que no las presentaban (HR=3.32, IC95%: 2.53 – 4.35 , P<0.001)
- Las complicaciones cardiacas está asociada al riesgo de morir (IC95% no contiene el 1 y P < 0.05)

Ejemplo: Regresión de Cox. Variable categórica

```
## Variable categórica
> quantile(xx$age, , right=F)
  0% 25% 50% 75% 100%
  30   59   72   82  104
> xx$age4gr <- cut( xx$age, breaks = c(quantile(xx$age)[1:4], 999) ,
+                      labels=1:4 , right=F )
> table(xx$age4gr)

  1   2   3   4
124 120 127 129
> sum(table(xx$age4gr))
[1] 500
```

- Se categoriza la variable edad utilizando los quartiles (59, 72 y 82) como puntos de corte
- Como se utiliza la opción *right=F*, los intervalos son [), cerrados a la izquierda, y esto da lugar a los 4 grupos:
 - Grupo 1 : menores de 59 años
 - Grupo 2 : 59 – 71
 - Grupo 3 : 72 – 81
 - Grupo 4 : mayores de 81

Ejemplo: Regresión de Cox. Variable categórica

```
> cox3 <- coxph ( Surv(lenfol, fstat) ~ age4gr, data=xx )
> summary(cox3)

      coef exp(coef)  se(coef)      z Pr(>|z|)    
age4gr2 0.7008    2.0155   0.3118  2.248   0.0246 *  
age4gr3 1.7483    5.7450   0.2786  6.276  3.48e-10 *** 
age4gr4 2.2746    9.7243   0.2728  8.339  < 2e-16 *** 

      exp(coef) exp(-coef) lower .95 upper .95    
age4gr2     2.015      0.4962   1.094    3.713    
age4gr3     5.745      0.1741   3.328    9.918    
age4gr4     9.724      0.1028   5.697   16.598    

Rsquare= 0.231  (max possible= 0.993 )
Likelihood ratio test= 131.2  on 3 df,  p=0
Wald test            = 103.9  on 3 df,  p=0
Score (logrank) test = 133.8  on 3 df,  p=0
```

- El riesgo de morir es 2 veces mayor en los individuos con edades entre 59 y 71 años con respecto a los menores de 59 años (HR=2.02, IC95%: 1.09 – 3.71 , P<0.001). Ese riesgo aumenta a 5.74 y 9.72 en los grupos de individuos entre 72 y 81, y mayores de 81 años, respectivamente (IC95%, P)
- La edad es un factor pronóstico para el riesgo de morir (LRT, P< 0.001)
- **Linealidad** en los coeficientes de regresión (0.70, 1.74, 2.27). Se debe incluir edad como variable continua

Ejercicios

- Fichero de datos: actg320.csv
 - Variable tiempo: “time”
 - Variable status: “censor”
1. Ajustar e interpretar el **modelo de Cox** para la variable binaria “tx”
 2. Ajustar e interpretar el **modelo de Cox** para la variable continua “cd4”
 3. Evaluar **la linealidad de la variable** “cd4”, categorizándola tomando como puntos de corte los cuartiles. Ajustar e interpretar **el modelo de Cox**

Variables de confusión

- Se desea analizar la relación entre la variable predictora X y la variable respuesta Y, pero hay una variable Z que puede afectar a la relación
- Z es una **variable de confusión o de control** en la asociación entre X e Y cuando
 - Z está relacionada con ambas, con X y con Y
 - Z no se encuentra en el camino causal entre X e Y
- Se ajustan 2 modelos de regresión: un modelo con X y otro modelo con X y Z
 - Si se ha producido un cambio en el coeficiente de regresión de X, entre **un 15% y un 25%**, diremos que Z está **confundiendo la relación** entre X e Y

Modelos de regresión múltiple

- En los modelos de regresión múltiples, **los coeficientes de regresión** quedan ajustados por la influencia que el resto de las variables tienen con la variable respuesta, y por las relaciones que existen entre las variables predictoras
- Los modelos de regresión múltiples son métodos más eficientes porque permiten estudiar efectos de varias variables **simultáneamente**
- Los modelos multivariantes de regresión de Cox deberían contener como máximo **1 variable por cada 10 eventos**

Interacción entre variables predictoras

- Existe una **interacción** entre las variables X y Z cuando la relación entre la variable predictora X y la variable respuesta es diferente en los niveles de una tercera variable Z
 - Al menos una de las variables de una interacción debe ser un factor pronóstico
 - Se dice que Z es una **variable modificadora** del efecto de X en Y
- El **modelo de interacción** entre X y Z se formula añadiendo un **término producto** de las 2 variables al modelo de efectos principales

$$h(t; X) = h_0(t) \cdot e^{\beta_1 X + \beta_2 Z + \beta_3 X \cdot Z}$$

- **Principio jerárquico:** el modelo debe contener los términos de orden inferior
- **Evaluación de la interacción** con el test $H_0: \beta_3=0$

Estrategias para la construcción de un modelo

- Paso 1: Análisis univariante. Regresión de Cox simple
- Paso 2: Modelo multivariante completo
 - Variables con $P < 0.25$ y las clínicamente relevantes
- Paso 3: Modelo multivariante preliminar de efectos principales
 - Proceso iterativo, evaluando conjuntamente variables estadísticamente significativas (Wald), variables de confusión y variables relevantes
- Paso 4: Evaluación de las variables continuas
 - Transformaciones. Categorización
- Paso 5: Interacciones

Ejemplo: Modelo Multivariante. Efectos principales

```
## Modelo Multivariante de Efectos Principales
> cox4 <- coxph ( Surv(lenfol,fstat) ~ age + hr + diasbp + bmi + gender + chf,
+                               data=xx)
> summary(cox4)

      coef exp(coef)   se(coef)      z Pr(>|z|)
age    0.049860  1.051123  0.006597  7.558 4.10e-14 ***
hr     0.011194  1.011257  0.002916  3.839 0.000123 ***
diasbp -0.010608  0.989448  0.003510 -3.022 0.002512 **
bmi    -0.045278  0.955731  0.016281 -2.781 0.005418 **
gender -0.271696  0.762086  0.143650 -1.891 0.058574 .
chf    0.779422  2.180213  0.146666  5.314 1.07e-07 ***
---
      exp(coef) exp(-coef) lower .95 upper .95
age    1.0511    0.9514   1.0376   1.0648
hr     1.0113    0.9889   1.0055   1.0171
diasbp 0.9894    1.0107   0.9827   0.9963
bmi    0.9557    1.0463   0.9257   0.9867
gender 0.7621    1.3122   0.5751   1.0099
chf    2.1802    0.4587   1.6355   2.9063

Rsquare= 0.34  (max possible= 0.993 )
Likelihood ratio test= 207.7  on 6 df,   p=0
Wald test            = 187.8  on 6 df,   p=0
Score (logrank) test = 207.1  on 6 df,   p=0
```

Ejemplo: Test del cociente de verosimilitudes

```
> ## Test del cociente de verosimilitudes, quitando "gebder"
> cox8 <- coxph ( Surv(lenfol,fstat) ~ age + hr + diasbp + bmi + chf, data=xx)
> anova(cox8, cox4, test="Chisq")
Analysis of Deviance Table

Cox model: response is Surv(lenfol, fstat)
Model 1: ~ age + hr + diasbp + bmi + chf
Model 2: ~ age + hr + diasbp + bmi + gender + chf
loglik Chisq Df P(>|Chi|)
1 -1125.2
2 -1123.5 3.58 1 0.05848 .
---
> drop1(cox4, test="Chisq")
Single term deletions

          Df     AIC     LRT   Pr(Chi)
<none> 2258.9
age      1 2321.0 64.094 1.186e-15 ***
hr       1 2271.1 14.150 0.0001688 ***
diasbp   1 2266.3  9.445 0.0021170 **
bmi      1 2264.9  7.984 0.0047205 **
gender   1 2260.5  3.580 0.0584801 .
chf      1 2284.9 28.028 1.195e-07 ***

```

- La función **anova()** compara las verosimilitudes de 2 modelos y **drop1()** compara el modelo de k variables con todos los modelos con k-1 variables

Ejemplo: Modelo Multivariante. Interacción

```
## Modelo Multivariante con Interacciones
> cox5 <- coxph ( Surv(lenfol,fstat) ~ age + hr + diasbp + bmi + gender + chf
+                               + age*gender, data=xx)
> summary(cox5)

            coef exp(coef)   se(coef)      z Pr(>|z|)
age        0.062450 1.064442 0.008377  7.455 8.96e-14 ***
hr         0.011250 1.011314 0.002908  3.869 0.000110 ***
diasbp    -0.010858 0.989200 0.003481 -3.119 0.001813 **
bmi        -0.043998 0.956955 0.016058 -2.740 0.006145 **
gender     2.256921 9.553625 0.960414  2.350 0.018776 *
chf        0.777736 2.176539 0.145622  5.341 9.25e-08 ***
age:gender -0.032134 0.968377 0.012121 -2.651 0.008022 **
---
            exp(coef) exp(-coef) lower .95 upper .95
age        1.0644     0.9395   1.0471    1.0821
hr         1.0113     0.9888   1.0056    1.0171
diasbp    0.9892     1.0109   0.9825    0.9960
bmi        0.9570     1.0450   0.9273    0.9876
gender     9.5536     0.1047   1.4543   62.7587
chf        2.1765     0.4594   1.6361    2.8955
age:gender 0.9684     1.0327   0.9456    0.9917

Rsquare= 0.349  (max possible= 0.993 )
Likelihood ratio test= 214.7  on 7 df,  p=0
Wald test             = 187.6  on 7 df,  p=0
Score (logrank) test = 208.5  on 7 df,  p=0
```

Ejemplo: Modelo Multivariante. Interacción

```
> B1 <- summary(cox5)$coef[1] ## age
> B2 <- summary(cox5)$coef[5] ## gender
> B3 <- summary(cox5)$coef[7] ## age*gender
>
> cox5$var
      [,1]          [,2]          [,3]          [,4]          [,5]          [,6]          [,7]
[1,] 7.016756e-05 1.606921e-07 4.861238e-06 3.464675e-05 4.851093e-03 -1.497942e-04 -6.178220e-05
[2,] 1.606921e-07 8.457157e-06 -2.476745e-06 1.471552e-07 1.406145e-05 -8.999136e-05 -6.449990e-07
[3,] 4.861238e-06 -2.476745e-06 1.211801e-05 -4.004409e-06 -2.332841e-05 2.191158e-05 6.779052e-07
[4,] 3.464675e-05 1.471552e-07 -4.004409e-06 2.578622e-04 6.605992e-04 -7.442045e-05 -6.229612e-06
[5,] 4.851093e-03 1.406145e-05 -2.332841e-05 6.605992e-04 9.223943e-01 -4.528261e-03 -1.151560e-02
[6,] -1.497942e-04 -8.999136e-05 2.191158e-05 -7.442045e-05 -4.528261e-03 2.120574e-02 2.527548e-05
[7,] -6.178220e-05 -6.449990e-07 6.779052e-07 -6.229612e-06 -1.151560e-02 2.527548e-05 1.469133e-04
>
> var_B2 <- cox5$var[5,5] ## gender
> var_B3 <- cox5$var[7,7] ## age*gender
> cov_B23 <- cox5$var[5,7] ## gender - age*gender
>
> age_set <- c(40,50,60,70,80,90)
>
> HR <- rep(NA,length(age_set))
> HR_L <- rep(NA,length(age_set))
> HR_U <- rep(NA,length(age_set))
```

- Se definen los niveles de edad en los que se está interesado: 40, 50, 60, 70, 80, 90
- Se extraen los valores de las varianzas y covarianzas de los estimadores de regresión

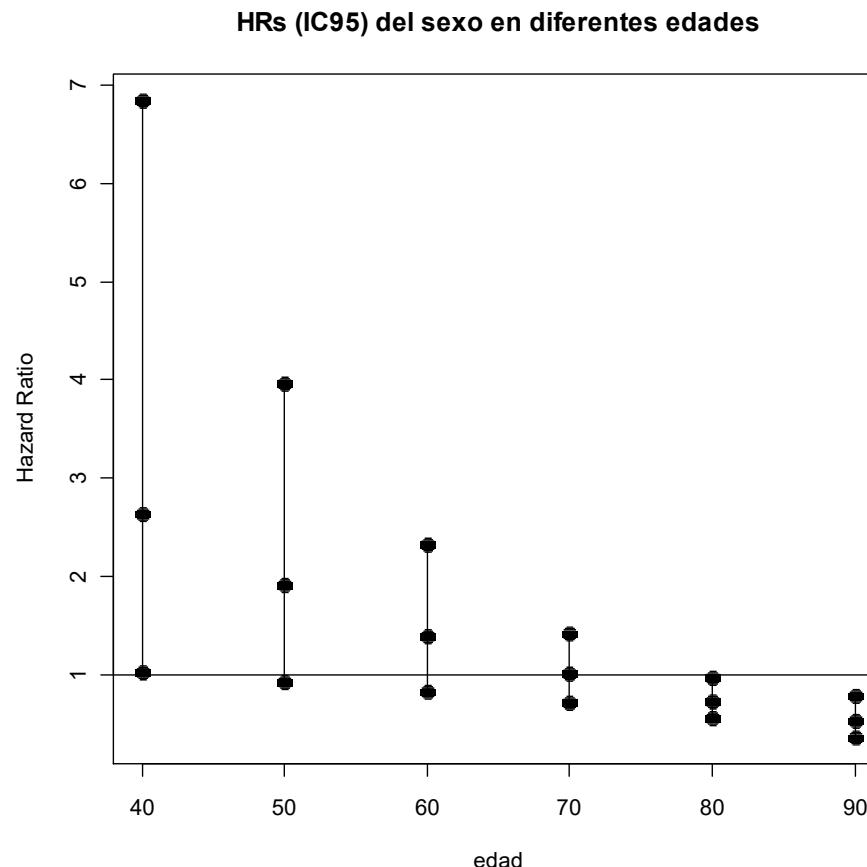
Ejemplo: Modelo Multivariante. Interacción

```
> i<-0
> for (x in age_set)
+ { i<-i+1
+   SE      <- sqrt(var_B2 + x^2 * var_B3 + 2*x *cov_B23)
+
+   HR[i]    <- exp( B2+x*B3)
+   HR_L[i]  <- exp( (B2+x*B3) - 1.96 * SE  )
+   HR_U[i]  <- exp( (B2+x*B3) + 1.96 * SE  )
+   print ( paste ( round( HR[i], dig=2) , " (",
+                  round( HR_L[i] , dig=2) , "-",
+                  round( HR_U[i] , dig=2) , ") ", sep ="" ) )
+ }
[1] "2.64 (1.02-6.85)"
[1] "1.92 (0.92-3.97)"
[1] "1.39 (0.83-2.33)"
[1] "1.01 (0.72-1.42)"
[1] "0.73 (0.55-0.97)"
[1] "0.53 (0.36-0.78)"
```

- El riesgo que tiene la variable sexo en el riesgo de morir tras un infarto, va decreciendo con la edad
- Con edades tempranas, el riesgo es mayor en mujeres, y en edades tardías el riesgo es mayor en hombres (ser mujer es protector)
- El riesgo es estadísticamente significativo cuando edad=40 y edad=80 y 90, donde el IC95% no contiene el 1

Ejemplo: Modelo Multivariante. Interacción

```
> dev.new()
> plot ( c(age_set, age_set, age_set), c(HR, HR_L, HR_U),
+         main="ORs (IC95) del sexo en diferentes edades",
+         xlab="edad", ylab="Odds Ratio", pch=16, cex= 1.5)
> for ( i in 1:length(HR))
+   {segments( age_set[i], HR_L[i], age_set[i], HR_U[i]) }
> abline(h=1)
```



Ejemplo: Análisis de influencia

```
> ## Likelihood Displacement  
> surv5 <- survreg(cox5, data=xx)  
> res_ld <- residuals(surv5, type="ldcase")  
> max(res_ld)  
[1] 0.2866523
```

- En regresión de Cox, el estadístico equivalente a la distancia de Cook se llama **Likelihood Displacement**
- Evalúa la influencia de las observaciones en el ajuste del modelo mediante una medida de **la diferencia entre las verosimilitudes** de los modelos con todas las observaciones y quitando cada una de ellas
- En nuestro ejemplo, no hay ninguna observación influyente en el ajuste del modelo de regresión de Cox

Ejemplo: Prediciendo supervivencia para individuos

```

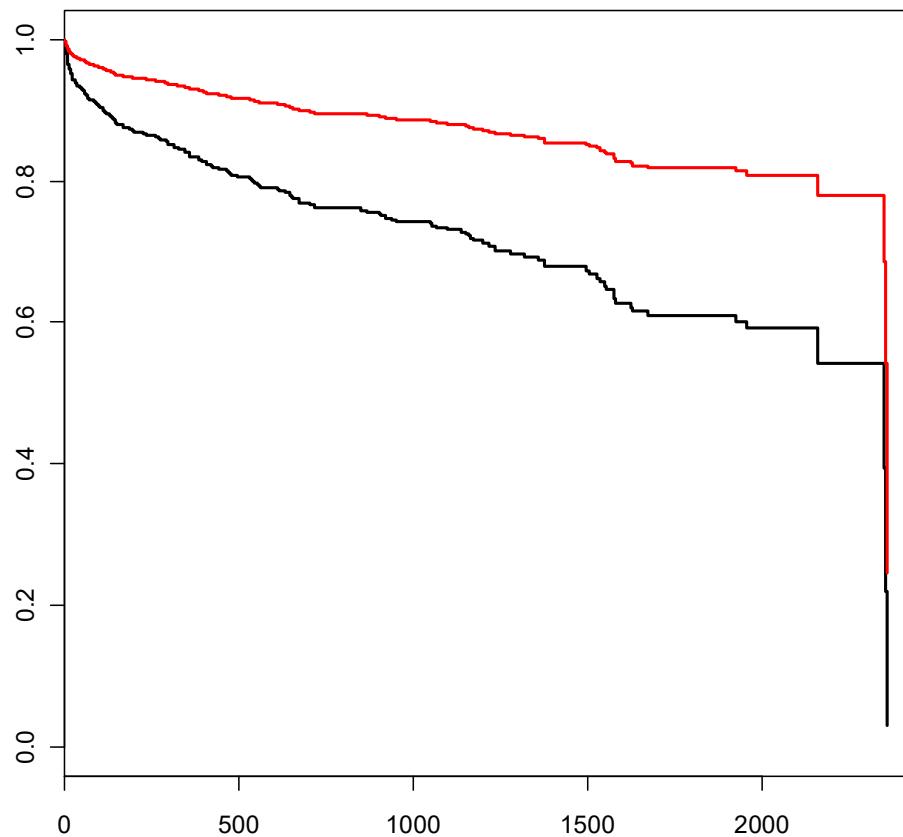
> ## Individuo con profile 1 (hombre de 70 años, línea negra)
> dev.new()
> surv_profile1 <- survfit ( cox5, newdata=data.frame(age = 70 , hr = 85 , diasbp = 80,
+                                bmi = 25, gender = 0 , chf = 0 ), se=F )
> plot( surv_profile1, mark.time=F, conf.int=F, col="black", lwd=2)
>
> ## Individuo con profile 2 (mujer de 40 años, línea roja)
> surv_profile2 <- survfit ( cox5, newdata=data.frame(age = 40 , hr = 85 , diasbp = 80,
+                                bmi = 25, gender = 1 , chf = 0 ), se=F )
> lines( surv_profile2, mark.time=F, conf.int=F, col="red", lwd=2 )
>
> summary(surv_profile2)
Call: survfit(formula = cox5, newdata = data.frame(age = 40, hr = 85,
diasbp = 80, bmi = 25, gender = 1, chf = 0), se.fit = F)

time n.risk n.event survival
 1    500      8   0.997
 2    492      8   0.994
 3    484      3   0.993
 4    481      2   0.992
 5    479      2   0.991
 6    477      5   0.989
 7    472      1   0.988
 8    471      1   0.987
 9    470      1   0.986
10    469      1   0.985
11    468      1   0.984
12    467      1   0.983
13    466      1   0.982
14    465      1   0.981
15    464      1   0.980
16    463      1   0.979
17    462      1   0.978
18    461      1   0.977
19    460      1   0.976
20    459      1   0.975
21    458      1   0.974
22    457      1   0.973
23    456      1   0.972
24    455      1   0.971
25    454      1   0.970
26    453      1   0.969
27    452      1   0.968
28    451      1   0.967
29    450      1   0.966
30    449      1   0.965
31    448      1   0.964
32    447      1   0.963
33    446      1   0.962
34    445      1   0.961
35    444      1   0.960
36    443      1   0.959
37    442      1   0.958
38    441      1   0.957
39    440      1   0.956
40    439      1   0.955
41    438      1   0.954
42    437      1   0.953
43    436      1   0.952
44    435      1   0.951
45    434      1   0.950
46    433      1   0.949
47    432      1   0.948
48    431      1   0.947
49    430      1   0.946
50    429      1   0.945
51    428      1   0.944
52    427      1   0.943
53    426      1   0.942
54    425      1   0.941
55    424      1   0.940
56    423      1   0.939
57    422      1   0.938
58    421      1   0.937
59    420      1   0.936
60    419      1   0.935
61    418      1   0.934
62    417      1   0.933
63    416      1   0.932
64    415      1   0.931
65    414      1   0.930
66    413      1   0.929
67    412      1   0.928
68    411      1   0.927
69    410      1   0.926
70    409      1   0.925
71    408      1   0.924
72    407      1   0.923
73    406      1   0.922
74    405      1   0.921
75    404      1   0.920
76    403      1   0.919
77    402      1   0.918
78    401      1   0.917
79    400      1   0.916
80    399      1   0.915
81    398      1   0.914
82    397      1   0.913
83    396      1   0.912
84    395      1   0.911
85    394      1   0.910
86    393      1   0.909
87    392      1   0.908
88    391      1   0.907
89    390      1   0.906
90    389      1   0.905
91    388      1   0.904
92    387      1   0.903
93    386      1   0.902
94    385      1   0.901
95    384      1   0.900
96    383      1   0.899
97    382      1   0.898
98    381      1   0.897
99    380      1   0.896
100   379      1   0.895
101   378      1   0.894
102   377      1   0.893
103   376      1   0.892
104   375      1   0.891
105   374      1   0.890
106   373      1   0.889
107   372      1   0.888
108   371      1   0.887
109   370      1   0.886
110   369      1   0.885
111   368      1   0.884
112   367      1   0.883
113   366      1   0.882
114   365      1   0.881
115   364      1   0.880
116   363      1   0.879
117   362      1   0.878
118   361      1   0.877
119   360      1   0.876
120   359      1   0.875
121   358      1   0.874
122   357      1   0.873
123   356      1   0.872
124   355      1   0.871
125   354      1   0.870
126   353      1   0.869
127   352      1   0.868
128   351      1   0.867
129   350      1   0.866
130   349      1   0.865
131   348      1   0.864
132   347      1   0.863
133   346      1   0.862
134   345      1   0.861
135   344      1   0.860
136   343      1   0.859
137   342      1   0.858
138   341      1   0.857
139   340      1   0.856
140   339      1   0.855
141   338      1   0.854
142   337      1   0.853
143   336      1   0.852
144   335      1   0.851
145   334      1   0.850
146   333      1   0.849
147   332      1   0.848
148   331      1   0.847
149   330      1   0.846
150   329      1   0.845
151   328      1   0.844
152   327      1   0.843
153   326      1   0.842
154   325      1   0.841
155   324      1   0.840
156   323      1   0.839
157   322      1   0.838
158   321      1   0.837
159   320      1   0.836
160   319      1   0.835
161   318      1   0.834
162   317      1   0.833
163   316      1   0.832
164   315      1   0.831
165   314      1   0.830
166   313      1   0.829
167   312      1   0.828
168   311      1   0.827
169   310      1   0.826
170   309      1   0.825
171   308      1   0.824
172   307      1   0.823
173   306      1   0.822
174   305      1   0.821
175   304      1   0.820
176   303      1   0.819
177   302      1   0.818
178   301      1   0.817
179   300      1   0.816
180   299      1   0.815
181   298      1   0.814
182   297      1   0.813
183   296      1   0.812
184   295      1   0.811
185   294      1   0.810
186   293      1   0.809
187   292      1   0.808
188   291      1   0.807
189   290      1   0.806
190   289      1   0.805
191   288      1   0.804
192   287      1   0.803
193   286      1   0.802
194   285      1   0.801
195   284      1   0.800
196   283      1   0.799
197   282      1   0.798
198   281      1   0.797
199   280      1   0.796
200   279      1   0.795
201   278      1   0.794
202   277      1   0.793
203   276      1   0.792
204   275      1   0.791
205   274      1   0.790
206   273      1   0.789
207   272      1   0.788
208   271      1   0.787
209   270      1   0.786
210   269      1   0.785
211   268      1   0.784
212   267      1   0.783
213   266      1   0.782
214   265      1   0.781
215   264      1   0.780
216   263      1   0.779
217   262      1   0.778
218   261      1   0.777
219   260      1   0.776
220   259      1   0.775
221   258      1   0.774
222   257      1   0.773
223   256      1   0.772
224   255      1   0.771
225   254      1   0.770
226   253      1   0.769
227   252      1   0.768
228   251      1   0.767
229   250      1   0.766
230   249      1   0.765
231   248      1   0.764
232   247      1   0.763
233   246      1   0.762
234   245      1   0.761
235   244      1   0.760
236   243      1   0.759
237   242      1   0.758
238   241      1   0.757
239   240      1   0.756
240   239      1   0.755
241   238      1   0.754
242   237      1   0.753
243   236      1   0.752
244   235      1   0.751
245   234      1   0.750
246   233      1   0.749
247   232      1   0.748
248   231      1   0.747
249   230      1   0.746
250   229      1   0.745
251   228      1   0.744
252   227      1   0.743
253   226      1   0.742
254   225      1   0.741
255   224      1   0.740
256   223      1   0.739
257   222      1   0.738
258   221      1   0.737
259   220      1   0.736
260   219      1   0.735
261   218      1   0.734
262   217      1   0.733
263   216      1   0.732
264   215      1   0.731
265   214      1   0.730
266   213      1   0.729
267   212      1   0.728
268   211      1   0.727
269   210      1   0.726
270   209      1   0.725
271   208      1   0.724
272   207      1   0.723
273   206      1   0.722
274   205      1   0.721
275   204      1   0.720
276   203      1   0.719
277   202      1   0.718
278   201      1   0.717
279   200      1   0.716
280   199      1   0.715
281   198      1   0.714
282   197      1   0.713
283   196      1   0.712
284   195      1   0.711
285   194      1   0.710
286   193      1   0.709
287   192      1   0.708
288   191      1   0.707
289   190      1   0.706
290   189      1   0.705
291   188      1   0.704
292   187      1   0.703
293   186      1   0.702
294   185      1   0.701
295   184      1   0.700
296   183      1   0.699
297   182      1   0.698
298   181      1   0.697
299   180      1   0.696
300   179      1   0.695
301   178      1   0.694
302   177      1   0.693
303   176      1   0.692
304   175      1   0.691
305   174      1   0.690
306   173      1   0.689
307   172      1   0.688
308   171      1   0.687
309   170      1   0.686
310   169      1   0.685
311   168      1   0.684
312   167      1   0.683
313   166      1   0.682
314   165      1   0.681
315   164      1   0.680
316   163      1   0.679
317   162      1   0.678
318   161      1   0.677
319   160      1   0.676
320   159      1   0.675
321   158      1   0.674
322   157      1   0.673
323   156      1   0.672
324   155      1   0.671
325   154      1   0.670
326   153      1   0.669
327   152      1   0.668
328   151      1   0.667
329   150      1   0.666
330   149      1   0.665
331   148      1   0.664
332   147      1   0.663
333   146      1   0.662
334   145      1   0.661
335   144      1   0.660
336   143      1   0.659
337   142      1   0.658
338   141      1   0.657
339   140      1   0.656
340   139      1   0.655
341   138      1   0.654
342   137      1   0.653
343   136      1   0.652
344   135      1   0.651
345   134      1   0.650
346   133      1   0.649
347   132      1   0.648
348   131      1   0.647
349   130      1   0.646
350   129      1   0.645
351   128      1   0.644
352   127      1   0.643
353   126      1   0.642
354   125      1   0.641
355   124      1   0.640
356   123      1   0.639
357   122      1   0.638
358   121      1   0.637
359   120      1   0.636
360   119      1   0.635
361   118      1   0.634
362   117      1   0.633
363   116      1   0.632
364   115      1   0.631
365   114      1   0.630
366   113      1   0.629
367   112      1   0.628
368   111      1   0.627
369   110      1   0.626
370   109      1   0.625
371   108      1   0.624
372   107      1   0.623
373   106      1   0.622
374   105      1   0.621
375   104      1   0.620
376   103      1   0.619
377   102      1   0.618
378   101      1   0.617
379   100      1   0.616
380   99      1   0.615
381   98      1   0.614
382   97      1   0.613
383   96      1   0.612
384   95      1   0.611
385   94      1   0.610
386   93      1   0.609
387   92      1   0.608
388   91      1   0.607
389   90      1   0.606
390   89      1   0.605
391   88      1   0.604
392   87      1   0.603
393   86      1   0.602
394   85      1   0.601
395   84      1   0.600
396   83      1   0.599
397   82      1   0.598
398   81      1   0.597
399   80      1   0.596
400   79      1   0.595
401   78      1   0.594
402   77      1   0.593
403   76      1   0.592
404   75      1   0.591
405   74      1   0.590
406   73      1   0.589
407   72      1   0.588
408   71      1   0.587
409   70      1   0.586
410   69      1   0.585
411   68      1   0.584
412   67      1   0.583
413   66      1   0.582
414   65      1   0.581
415   64      1   0.580
416   63      1   0.579
417   62      1   0.578
418   61      1   0.577
419   60      1   0.576
420   59      1   0.575
421   58      1   0.574
422   57      1   0.573
423   56      1   0.572
424   55      1   0.571
425   54      1   0.570
426   53      1   0.569
427   52      1   0.568
428   51      1   0.567
429   50      1   0.566
430   49      1   0.565
431   48      1   0.564
432   47      1   0.563
433   46      1   0.562
434   45      1   0.561
435   44      1   0.560
436   43      1   0.559
437   42      1   0.558
438   41      1   0.557
439   40      1   0.556
440   39      1   0.555
441   38      1   0.554
442   37      1   0.553
443   36      1   0.552
444   35      1   0.551
445   34      1   0.550
446   33      1   0.549
447   32      1   0.548
448   31      1   0.547
449   30      1   0.546
450   29      1   0.545
451   28      1   0.544
452   27      1   0.543
453   26      1   0.542
454   25      1   0.541
455   24      1   0.540
456   23      1   0.539
457   22      1   0.538
458   21      1   0.537
459   20      1   0.536
460   19      1   0.535
461   18      1   0.534
462   17      1   0.533
463   16      1   0.532
464   15      1   0.531
465   14      1   0.530
466   13      1   0.529
467   12      1   0.528
468   11      1   0.527
469   10      1   0.526
470   9      1   0.525
471   8      1   0.524
472   7      1   0.523
473   6      1   0.522
474   5      1   0.521
475   4      1   0.520
476   3      1   0.519
477   2      1   0.518
478   1      1   0.517
479   0      1   0.516

```

- Se comparan las curvas de supervivencia predichas por el modelo de Cox de un hombre de 70 años con una mujer de 40, ambos con bmi=25, hr=85, diasbp=80, chf=0

Ejemplo: Prediciendo supervivencia para individuos



- La supervivencia de la mujer de 40 años (rojo) es claramente mejor que la del hombre de 70 (negro)
- Se observa que las curvas son “paralelas”: la distancia aumenta gradualmente, los escalones se producen en los mismos tiempos (principio de riesgos proporcionales)

Modelo de Cox estratificado

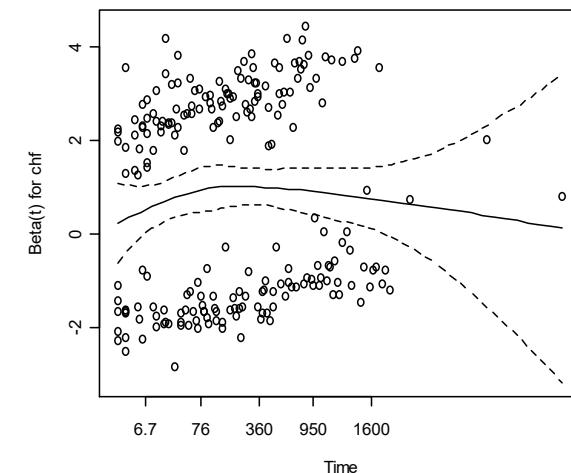
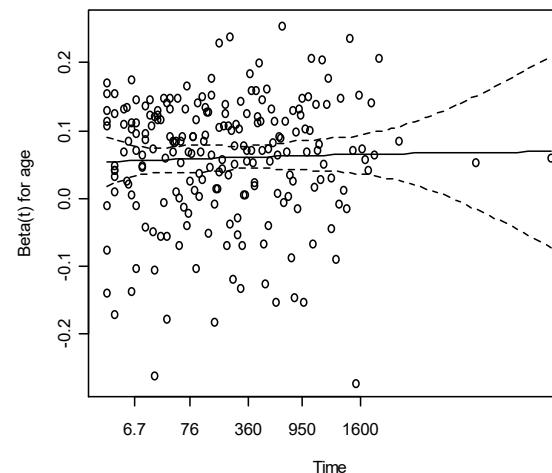
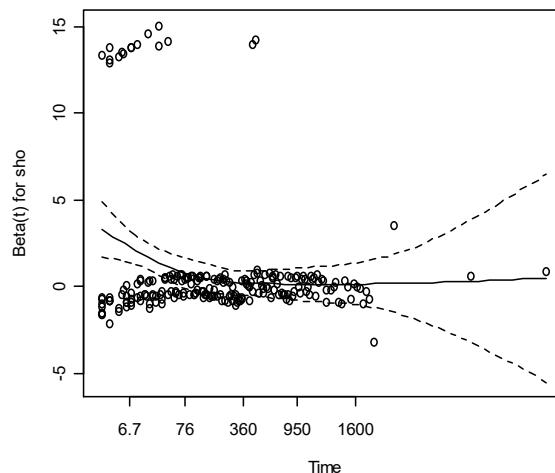
- Si una de las variables no cumple el supuesto de riesgos proporcionales, la alternativa es ajustar un **modelo de Cox estratificado**, donde se ajusta una función de riesgo de referencia (baseline) para cada categoría de la variable que forma los **estratos**
 - Si la variable que no cumple el supuesto es continua, hay que categorizarla

$$h_j(t; X) = h_{0j}(t) \cdot e^{\beta_1 X_1 + \dots + \beta_p X_p} \quad j=1, \dots, J \text{ estratos o categorías de la variable que no cumple el supuesto}$$

- El modelo supone que las variables predictoras tienen el **mismo efecto en todos los estratos**. Se sigue estimando un solo coeficiente por variable
 - Se pueden incluir términos de interacción para validarla y corregir los efectos
- Una situación común donde se usa el modelo estratificado es con variables donde es difícil asumir que los riesgos de referencia son iguales
 - Por ejemplo, el hospital en un estudio multicéntrico

Ejemplo: Modelo de Cox estratificado

```
> cox10 <- coxph ( Surv(lenfol, fstat) ~ sho + age + chf, data=xx )
> summary(cox10)
      coef exp(coef) se(coef)     z Pr(>|z|)
sho 0.892605  2.441482 0.261365 3.415 0.000637 ***
age 0.059348  1.061145 0.006169 9.621 < 2e-16 ***
chf 0.821392  2.273663 0.143303 5.732 9.93e-09 ***
> cox.zph(cox10)
      rho chisq      p
sho -0.2093 8.966 0.00275
age  0.0357 0.280 0.59638
chf  0.0442 0.448 0.50330
GLOBAL      NA 9.640 0.02189
> dev.new(); par (mfrow=c(1,3))
> for ( i in 1:3 ) plot(cox.zph(cox10), var=i)
```



- Se rechaza la hipótesis de cumplimiento del supuesto de riesgos proporcionales para la variable “sho”. En el gráfico se ve que el coeficiente cambia a lo largo del tiempo

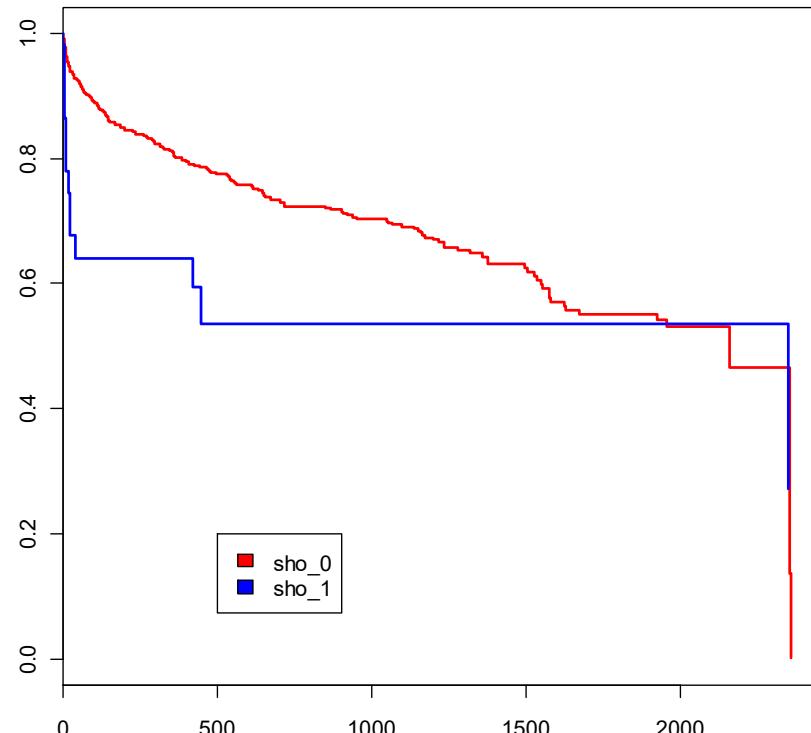
Ejemplo: Modelo de Cox estratificado

```
> cox11 <- coxph ( Surv(lenfol, fstat) ~ strata(sho) + age + chf, data=xx )
> summary(cox11)

      coef exp(coef)  se(coef)      z Pr(>|z|)
age 0.058298  1.060031 0.006163 9.460  < 2e-16 ***
chf 0.817181  2.264108 0.144458 5.657 1.54e-08 ***
---
<--->

> dev.new()
> plot (survfit ( cox11 ), col=c("red","blue"), mark.time=F, lwd=2 )
> legend ( 500, 0.2 , legend=c("sho_0","sho_1"), c("red","blue"))
```

- Se ajusta un modelo estratificado con *strata(sho)*
- No se estima un coeficiente para “sho”. La estimación del riesgo de “age” y “chf” es parecida
- La supervivencia es distinta para los que han sufrido un shock cardiaco, que mueren muy rápidamente



Ejemplo: Modelo de Cox estratificado

```
> ## Modelo de Cox estratificado, que incluye las interacciones
> cox12 <- coxph ( Surv(lenfol, fstat) ~ strata(sho) + age + chf +
+                               sho:age + sho:chf, data=xx )
> summary(cox12)

            coef exp(coef)    se(coef)      z Pr(>|z|)
age        0.058992  1.060767  0.006295  9.371 < 2e-16 ***
chf        0.831325  2.296358  0.148163  5.611 2.01e-08 ***
age:sho -0.017737  0.982420  0.029820 -0.595   0.552
chf:sho -0.252614  0.776768  0.614869 -0.411   0.681
```

- El modelo de Cox estratificado asume que el efecto de las variables predictoras es igual en cada estrato
- Para validar este supuesto y corregir los efectos en cada estrato, se pueden introducir en el modelo **interacciones** de las variables predictoras con la variable que define los estratos
- En este caso las 2 interacciones no son significativas, y por tanto, podemos asumir que **los efectos de las variables** “age” y “chf” en la función de riesgo **son iguales** en los 2 estratos definidos por la variable “sho”
- Las estimaciones de los efectos de las variables “age” y “chf” han cambiado poco

Ejercicios

- Fichero de datos: actg320.csv
 - Variable tiempo: “time”
 - Variable status: “censor”
1. Ajustar e interpretar el **modelo de Cox multivariante** con las variables:
 - tx + age + sex + cd4 + priorzdv
 2. Ajustar e interpretar el modelo de Cox multivariante **de efectos principales**
 - Eliminando algunas variables que no son estadísticamente significativas
 3. Ajustar e interpretar el modelo de Cox multivariante con **interacciones**
 4. Comprobar si el modelo final cumple el supuesto de **riesgos proporcionales**

Estadística Aplicada a la Investigación Biomédica con R

20 Modelos Predictivos. Curvas ROC

- ✓ **Regresión Logística. Clasificación y Predicción**
 - ✓ **Sensibilidad y Especificidad**
 - ✓ **Análisis de curvas ROC**
-
- ✓ **Regresión de Cox. Predicción y Risk score**
 - ✓ **Regresión de Cox. C-index. Curvas ROC**

Regresión Logística. Predicción y Clasificación

- El modelo de regresión logística ajustado proporciona **probabilidades estimadas** de que ocurra el suceso de interés para cada individuo, y estas probabilidades pueden utilizarse para **clasificar** a los individuos
- Si se fija un **punto de corte** P_0 entre 0 y 1, podemos clasificar a los individuos en dos grupos:
 - Si $p_i > P_0$ el modelo predice que el suceso está **presente** y clasificamos al individuo como **1**
 - Si $p_i < P_0$ el modelo predice que el suceso está **ausente** y clasificamos al individuo como **0**
 - Dependiendo del valor de P_0 la clasificación será mejor o peor
 - Un valor habitual es 0.5
 - Otra opción es tomar P_0 como la **proporción de individuos que presentan el suceso** en la muestra. Tiene sentido ya que la suma de las probabilidades estimadas es igual al número de individuos que presentan el suceso

Tabla de clasificación

- A partir de un determinado **punto de corte P_0** , se genera la **tabla de clasificación** o tabla de confusión

		Predicho		Total	
Observado			Total		
	$Y=1$	$Y=0$			
$Y=1$	VP	FN	VP + FN		
$Y=0$	FP	VN	FP + VN		
Total	VP + FP	FN + VN			

- **Verdaderos positivos (VP)**: individuos con presencia del suceso y predicción correcta
- **Verdaderos negativos (VN)**: individuos con ausencia del suceso y predicción correcta
- **Falsos positivos (FP)**: individuos con ausencia del suceso y predicción incorrecta
- **Falsos negativos (FN)**: individuos con presencia del suceso y predicción incorrecta

Sensibilidad y Especificidad

- **Sensibilidad:** es la capacidad del clasificador para detectar los **verdaderos positivos (VP)**
 - Proporción de positivos detectados por el modelo entre los individuos que presentaban el suceso

$$\text{Sensibilidad} = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

- **Especificidad:** es la capacidad del clasificador para detectar los **verdaderos negativos (VN)**
 - Proporción de negativos detectados por el modelo entre los individuos que no presentaban el suceso

$$\text{Especificidad} = \frac{\text{VN}}{\text{VN} + \text{FP}}$$

Índices de Exactitud

- **Precisión, Exactitud (Accuracy)**
 - Probabilidad de predecir correctamente
 - Clasificador ideal: $AC=1$
 - Coste similar para FN y FP

$$AC = \frac{VP + VN}{N}$$

- **Índice de Youden**
 - Basado en la sensibilidad y especificidad
 - Clasificador ideal: $\gamma=1$

$$\gamma = S + E - 1$$

Curvas ROC

- **Curvas ROC** (Receiver Operating Characteristics)
 - La sensibilidad y especificidad están referidos a un único punto P_0 , y la metodología de las curvas ROC va enfocada al análisis del **rendimiento global** de un clasificador
 - Las curvas ROC proporcionan una descripción más completa de la **capacidad predictiva** de un modelo
 - Las curvas ROC permiten la **comparación entre modelos** para saber cuál predice mejor
- Se utiliza para evaluar **pruebas diagnósticas** y puede ser usada en el contexto de **predicción y clasificación de la regresión logística**
- Las curvas ROC se han incorporado como método de comparación de la **capacidad predictiva** entre técnicas de clasificación en el contexto de las técnicas de **data mining**

Curvas ROC

- La curva ROC se define con los puntos (**1 – especificidad, sensibilidad**)
 - Se calcula la sensibilidad y especificidad **para todos** los valores posibles del punto de corte P_0
 - Se muestran en **un gráfico** la fracción de VPs (sensibilidad) frente a la fracción de FPs (1-especificidad)
- El clasificador con máxima sensibilidad y especificidad se obtiene cuando la curva ROC se acerque al **punto (0,1)**
- La curva ROC permite la **selección de puntos de corte** en el clasificador con buena sensibilidad y/o especificidad

AUC. Área bajo la curva ROC

- El **área bajo la curva ROC (AUC)** es una medida de la **capacidad predictiva global del modelo**
- Se usa para **comparar modelos predictivos**
 - Según el AUC, los clasificadores se consideran
 - $AUC = 0.5$ el modelo clasifica de forma aleatoria
 - $AUC > 0.70 - 0.75$ buen poder predictivo
 - $AUC > 0.90 - 0.95$ excelente poder predictivo
 - $AUC = 1$ (máximo) clasificador ideal
- El área bajo la curva ROC (AUC) se puede **interpretar** como un **índice de concordancia**, como la probabilidad de que el modelo asigne un valor mayor en las observaciones de la clase 1 que a las de la clase 0
 - AUC tiene relación con el test no paramétrico de suma de rangos de Wilcoxon

Ejemplo: Capacidad predictiva. Curvas ROC

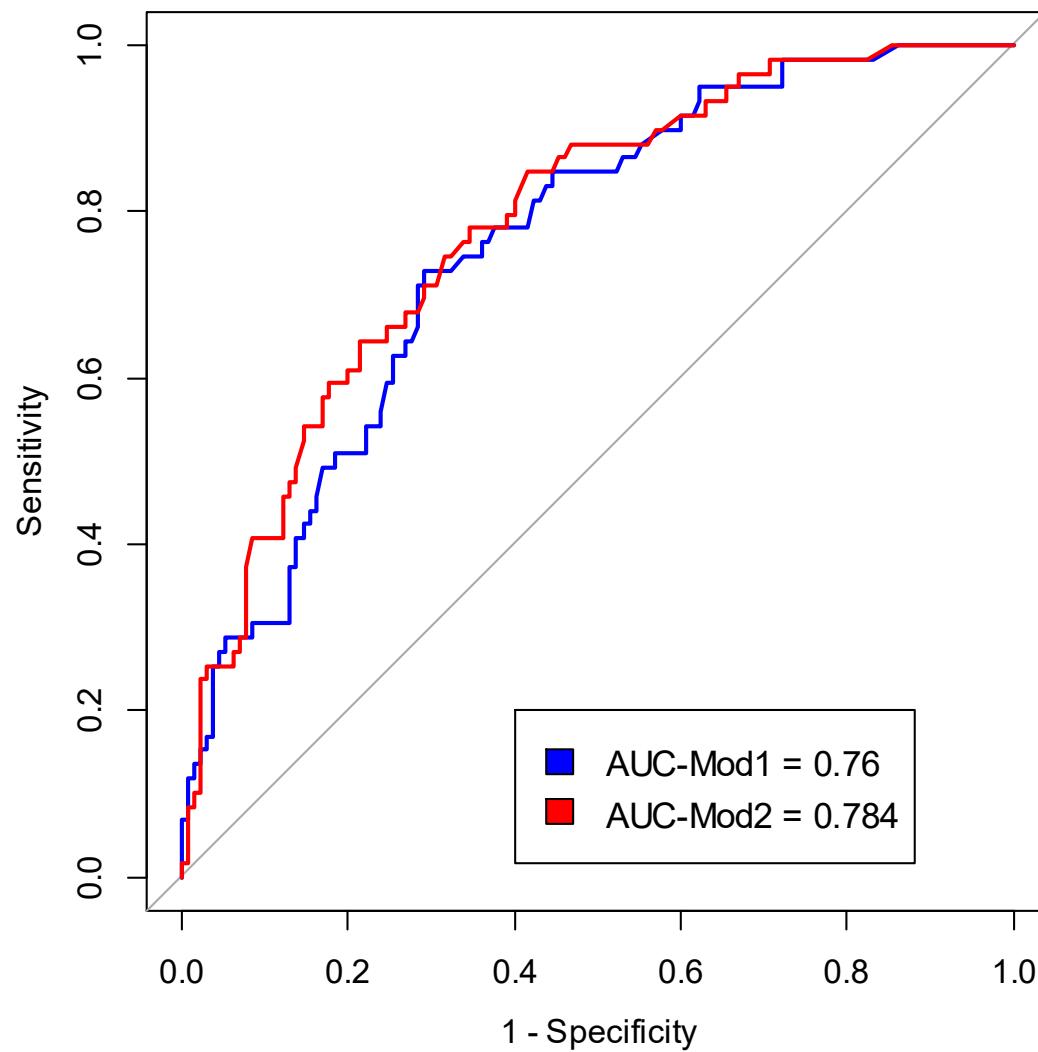
```
> library(gplots)
> library(pROC)
> ## Fichero Datos: Bajo peso al nacer
> xx <- read.csv(file="C:/Bioestadistica con R/Datos/Bajo peso al nacer.csv", sep=";")
>
> out.7 <- glm( bajo_pes ~ edad + peso_gr + raza + fumador + hta + irr_urin + part_pre2
+ , data=xx, family = binomial )
>
> out.8 <- glm( bajo_pes ~ edad + peso_gr + raza + fumador + hta + irr_urin + part_pre2
+ edad*peso_gr + peso_gr*fumador , data=xx, family = binomial )
>
> ## Probabilidades ajustadas por el modelo
> prob.7 <- predict.glm ( out.7, type="response")
> prob.8 <- predict.glm ( out.8, type="response")
>
> ## Análisis de la Curva ROC
> roc.out.7 <- roc( xx$bajo_pes , prob.7 )
> roc.out.8 <- roc( xx$bajo_pes , prob.8 )
```

- Se va a comparar **la capacidad predictiva** de 2 modelos de regresión logística, que se diferencia en un término de la interacción
- Se utiliza la función ***roc()*** del paquete ***pROC***
- Hay otro paquete, **ROCR**, que se usa también bastante

Ejemplo: Capacidad predictiva. AUC

```
> ## AUC con IC95%
> out.auc.7 = auc (roc.out.7)
> out.auc.7
Area under the curve: 0.7596
> w.auc.7 = out.auc.7 [1]
> w.auc.7
[1] 0.759648
>
> out.auc.8 = auc (roc.out.8)
> w.auc.8 = out.auc.8 [1]
> w.auc.8
[1] 0.7842894
> ## IC95%
> out.ci.7 = ci.auc (roc.out.7)
> out.ci.7
95% CI: 0.6891-0.8302 (DeLong)
> out.ci.8 = ci.auc (roc.out.8)
> out.ci.8
95% CI: 0.7163-0.8523 (DeLong)
>
> ## Gráfico de la curva ROC
> dev.new()
> plot( roc.out.7 , legacy.axes = TRUE, col="red" )
> plot( roc.out.8 , legacy.axes = TRUE, col="blue", add=T )
> ## Leyenda del gráfico
> label.7 <- paste("AUC-Mod1 =", round(w.auc.7, dig=3))
> label.8 <- paste("AUC-Mod2 =", round(w.auc.8, dig=3))
> legend(0.6,0.2, c(label.7, label.8), c("blue","red"), cex = 1.1
```

Ejemplo: Capacidad predictiva. Curvas ROC



- El modelo 2 predice mejor que el modelo 1

Ejemplo: Capacidad predictiva. Índice de Youden

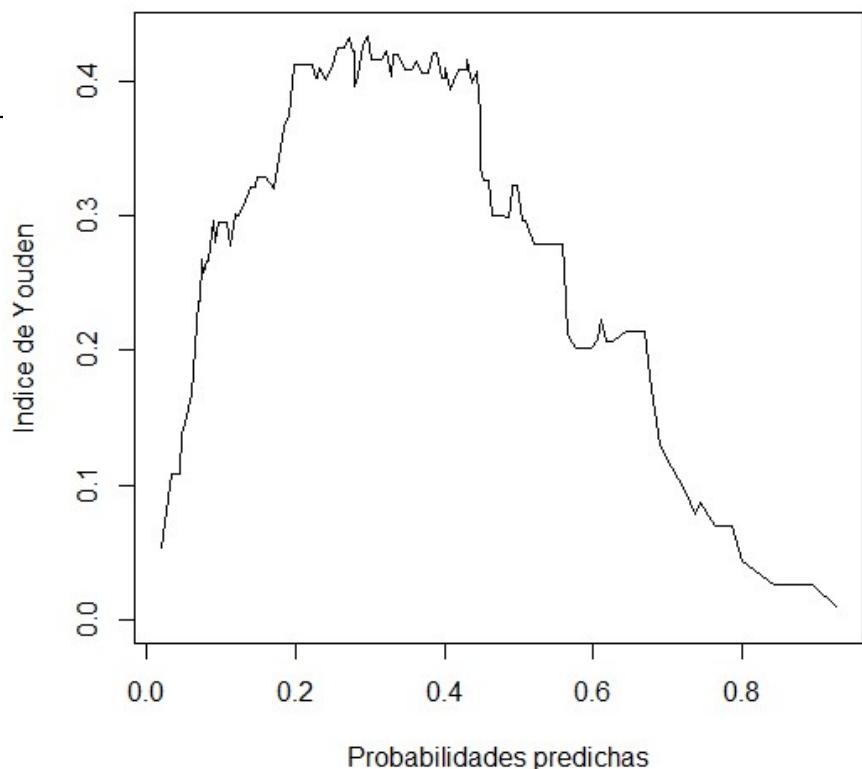
```
> ## Extraer la Sensibilidad y Especificidad
> espe <- coords ( roc.out.8, roc.out.8$thresholds, "thr", "sp")
> sens <- coords ( roc.out.8, roc.out.8$thresholds, "thr", "se")
> class(sens)
[1] "matrix"
> dim(sens)
[1] 1 129
> sens[1,1:12]
      -Inf  0.020921410214854  0.0337673698278218  0.040845420572457
      1.0000000  1.0000000  1.0000000  1.0000000
0.0442650419749042  0.0479566213976923  0.0519393118347498  0.0612132028187423
      1.0000000  1.0000000  1.0000000  1.0000000
0.0702604027197041  0.073035706655882  0.0741040695058444  0.077085201218838
      1.0000000  1.0000000  0.9830508  0.9830508
>
> ## Índice de Youden
> ind.youden <- sens + espe - 1
> dev.new()
> plot( roc.out.8$thresholds, ind.youden, type="l"
+       , xlab="Probabilidades predichas", ylab="Índice de Youden")
>
```

- Se guardan la sensibilidad y especificidad para todos los thresholds posibles
- Se crea el índice de Youden = sens + espec – 1
- Se representa frente los thresholds

Ejemplo: Capacidad predictiva. Índice de Youden

```
> ## Maximo del Indice de Youden
> max(ind.youden)
[1] 0.4335072
> ## Posicion donde está el máximo
> which.max(ind.youden)
[1] 55
> ## Probabilidad predicha que da el máximo
> roc.out.8$thresholds[55]
[1] 0.2839115
> sens[55]
[1] 0.779661
> espe[55]
[1] 0.6538462
```

- El **punto de corte en la probabilidad** que obtiene mejor índice de Youden es 0.284
- En ese punto, la **sensibilidad** es 77.97% y la **especificidad** es 65.38%
- El rango entre probabilidades 0.25 y 0.45 da valores parecidos al máximo del índice de Youden



Regresión de Cox. Predicción y risk score

- El modelo de regresión de Cox ajustado proporciona **estimaciones del riesgo y de la supervivencia** para cada individuo
 - Se pueden utilizar para **clasificar** a los individuos
- El **predictor lineal** en el modelo de regresión de Cox, se puede usar como las probabilidades en regresión logística. Se le llama **risk score**

$$h(t; X) = h_0(t) \cdot e^{b_1 X_1 + \dots + b_p X_p}$$

$$\text{Risk Score} = \text{Pred. Lin.} = b_1 X_1 + \dots + b_p X_p$$

- Como la función exponencial es monótona creciente, individuos con altos valores en el **risk score**, tendrán **alto riesgo** de que se produzca el evento estudiado

Regresión de Cox. Predicción y risk score

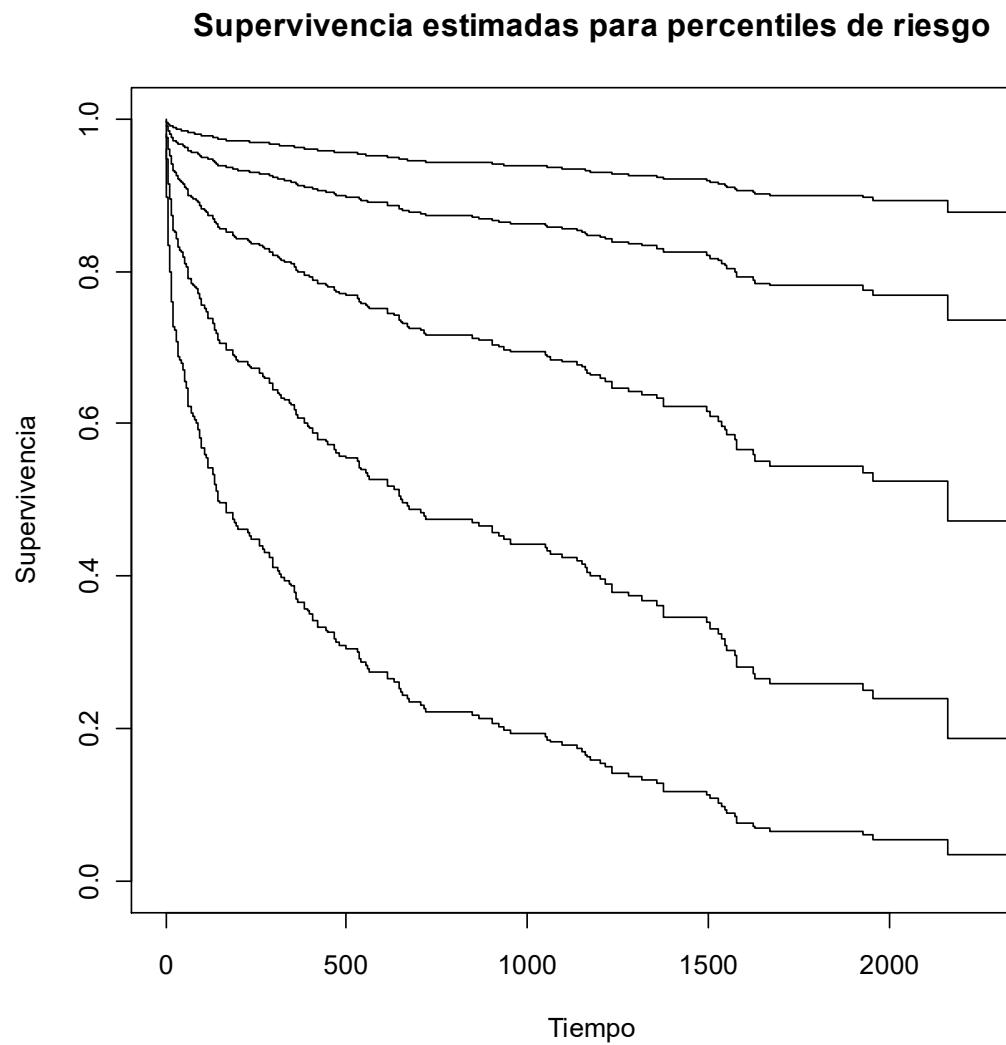
- Representación gráfica de los **percentiles de riesgo**
 - Se suelen calculan los percentiles 10, 25, 50, 75, 90 del risk score
- Para estimar la función de **supervivencia** en esos percentiles r_q
 - Supervivencia baseline
 - $\exp(r_q)$ es la exponencial del valor del percentil en el risk score

$$\hat{S}(t_i, r_q) = [\hat{S}_0(t_i)]^{\exp(r_q)} \quad i = 1, 2, \dots, n$$

Ejemplo: Percentiles de riesgo

```
> cox5 <- coxph ( Surv(lenfol,fstat) ~ age + hr + diasbp + bmi + gender + chf
+                               + age*gender, data=xx)
> ## Definición del Risk Score (predictor lineal)
> risk.score <- cox5$linear.predictors
>
> ## Percentiles de riesgo 0.10, 0.25, 0.5, 0.75, 0.90
> risk_quant<-quantile(risk.score, c(0.10, 0.25, 0.5, 0.75, 0.90 ))
> risk_quant
    10%      25%      50%      75%      90%
-1.68160865 -0.83875709  0.05847611  0.85966227  1.55989171
>
> ## Cálculo de la supervivencia
>
> S0t<-exp(-basehaz(cox5)[1])
> times<-basehaz(cox5)[2]
>
> ## Gráfica de los percentiles de riesgo
> dev.new()
> plot(times[,1] , S0t[,1]**exp(risk_quant[1]),type="s",
+       main="Supervivencia estimadas para percentiles de riesgo",
+       xlab="Tiempo", ylab="Supervivencia" ,ylim=c(0,1))
> lines(times[,1] , S0t[,1]**exp(risk_quant[2]),type="s")
> lines(times[,1] , S0t[,1]**exp(risk_quant[3]),type="s")
> lines(times[,1] , S0t[,1]**exp(risk_quant[4]),type="s")
> lines(times[,1] , S0t[,1]**exp(risk_quant[5]),type="s")
>
```

Ejemplo: Percentiles de riesgo



C-index

- El **C-index** es una medida de concordancia que se usa para evaluar la **capacidad predictiva** de un **modelo de supervivencia**
 - Es una **generalización del AUC** para datos de supervivencia
- El **C-index** se define como la proporción de pares de observaciones para los cuales el orden de los tiempos de **supervivencia observados** y las **predicciones** del modelo son **concordantes**
 - Se omiten aquellos pares en los que el tiempo más corto es censurado
- **Observación:** las funciones que calculan el **c-index** comparan supervivencias observadas con supervivencias predichas. En los casos en los que se tiene un **risk score**, se puede utilizar con **signo negativo**

C-index

- Se comparan todos los pares posibles de individuos. Se podrían establecer como máximo $N^*(N-1)/2$ comparaciones, si no hubiera censuras
 - Se omiten aquellos pares en los que el tiempo más corto es censurado
- El **c-index** es el número de concordancias entre lo observado y lo predicho, dividido por el número de comparaciones que se han establecido

Caso	Observado (tiempo y evento)	Predicho RS(A) > RS(B)	Predicho RS(B) > RS(A)
1	Ind A ——x	Concordancia	Discordancia
	Ind B —————x		
2	Ind A ——x	Concordancia	Discordancia
	Ind B —————●		
3	Ind A —●	??	??
	Ind B —————x		
4	Ind A —●	??	??
	Ind B —————●		

x – evento
o – censurado

RS = risk score

Regresión de Cox. Análisis de curvas ROC

- El **análisis de curvas ROC** se ha extendido al análisis de supervivencia. Para ello es necesario **fijar t** un tiempo de supervivencia de interés. Este análisis se conoce como **time-dependent ROC**
- Se evalúa la capacidad predictiva del modelo con el **AUC de la curva ROC** para predecir el evento en el **instante t** fijado. Se denota como **AUC(t)**
- En análisis de supervivencia, se basa en la definición de una **variable binaria cambiante en el tiempo**, que tomará valor 1 si el evento ha ocurrido antes del instante t

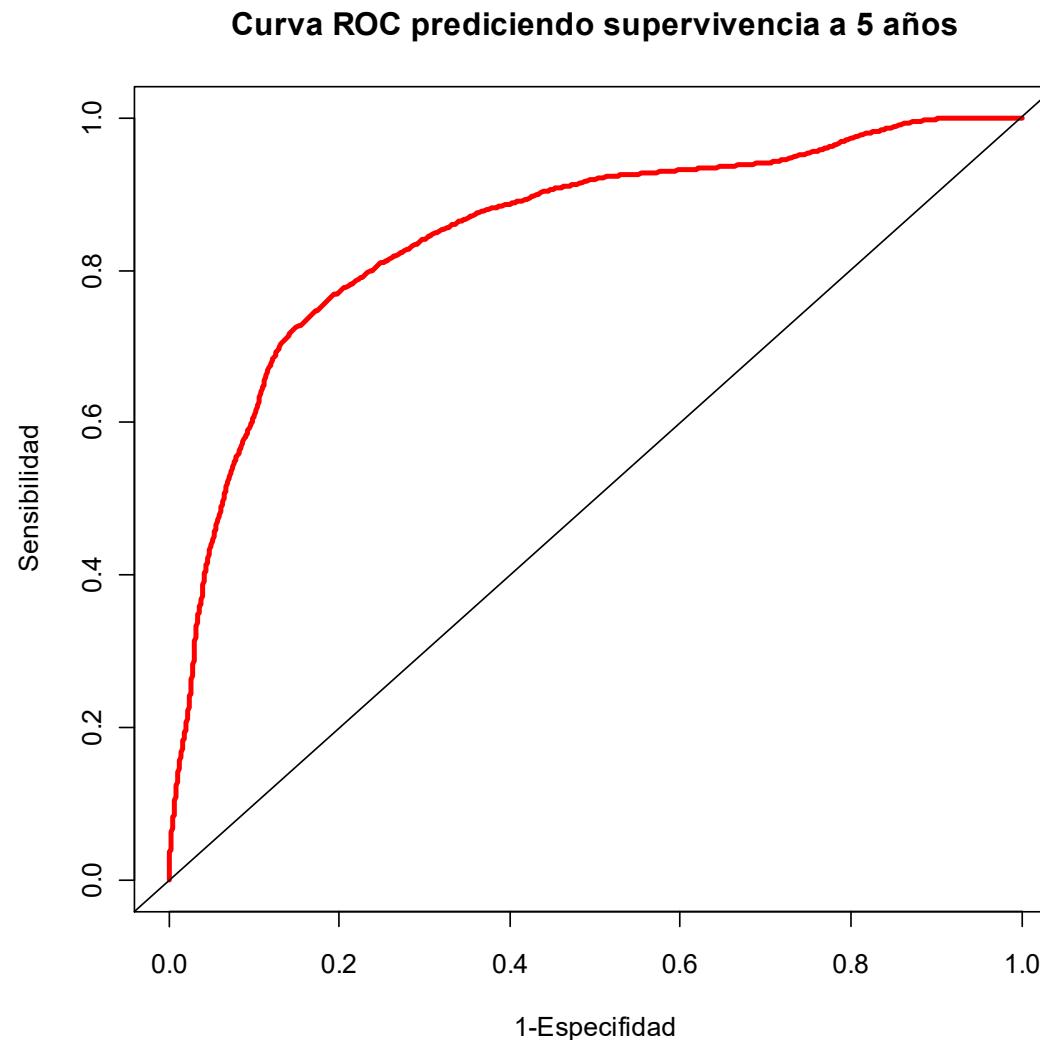
$$D_i(t) = \begin{cases} 1 & \text{si } T_i \leq t \\ 0 & \text{si } T_i > t \end{cases}$$

- Se adaptan las definiciones de **sensibilidad** y **especificidad** en función del tiempo, teniendo en cuenta las observaciones censuradas. Se utilizan métodos **no paramétricos**

Ejemplo: Curvas ROC. Análisis de supervivencia

```
> library(survivalROC)
> library(rms)
> ## C-index (con el risk score cambiado de signo)
> rcorr.cens ( - risk.score , Surv(xx$lenfol,xx$fstat) ) ["C Index"]
  C Index
0.7784535
>
> ## Análisis de curva ROC
>
> nobs <- nrow(xx)
> cutoff <- 5 * 365.25 ## 5 AÑOS
>
> surv.ROC5 = survivalROC( xx$lenfol, xx$fstat, risk.score,
+                           predict.time = cutoff, span = 0.25*nobs^(-0.20) )
>
> ## Curva ROC
> dev.new()
> plot(surv.ROC5$FP, surv.ROC5$TP, type="l", xlim=c(0,1), ylim=c(0,1),
+       col="red", lwd=3, xlab="1-Especificidad", ylab="Sensibilidad",
+       main="Curva ROC prediciendo supervivencia a 5 años")
>
> abline(0,1)
>
> ## AUC
> surv.ROC5$AUC
[1] 0.8393066
```

Ejemplo: Curvas ROC. Análisis de supervivencia



Ejercicio

- Fichero de datos: umaru.txt
 - variable respuesta: DFREE
- Estudiar mediante el análisis de **la curva ROC** la capacidad predictiva del modelo de regresión logística de efectos principales
 - AGE + IVHX + NDRUGTX + RACE + TREAT + SITE

Estadística Aplicada a la Investigación Biomédica con R

21 Análisis de Medidas Repetidas

- ✓ **Análisis de medidas repetidas**
- ✓ **Modelos lineales mixtos**
- ✓ **Test de Friedman**

Análisis de medidas repetidas

- El **análisis de medidas repetidas** es una técnica estadística que se usa en los estudios en los que una variable es medida **varias veces** en un conjunto de individuos
 - El análisis incluye un **factor intra-sujeto** y es una generalización de los estudios pareados
 - Se controla el **efecto** debido a las **diferencias entre individuos**
- Se conocen también como **análisis de datos longitudinales** ya que la variable respuesta se mide en **varios instantes del tiempo** para cada individuo en el estudio
- Algunas **complicaciones** de este diseño:
 - **Dependencia** entre las medidas repetidas hechas en la misma unidad experimental
 - Pueden presentar un porcentaje alto de **datos perdidos**

Análisis de medidas repetidas

- El análisis de medidas repetidas se puede usar en general cuando se evalúa una variable respuesta en un conjunto de individuos bajo **diferentes condiciones experimentales**
 - Algunos diseños especiales en **ensayos clínicos**
 - Diferentes tratamientos aplicados al mismo individuo
 - También si las observaciones están agrupadas en clusters o grupos (diseños anidados)
- El modelo más sencillo es el **análisis de medidas repetidas de un factor** que incluye una única variable respuesta medida en **J niveles** en los mismos individuos
 - Por ejemplo: J períodos del tiempo o J condiciones experimentales
 - Todos los individuos de la muestra pasan por todos los niveles del factor

Modelos lineales mixtos

- Un **modelo lineal mixto** es una generalización del modelo lineal general
 - **Efectos fijos y aleatorios**
 - **Factor fijo:** los niveles experimentales son todos los posibles niveles, y son fijados por el investigador
 - **Factor aleatorio:** los niveles son una muestra de todos los posibles niveles. Los niveles actúan como clusters
- Se pueden usar para analizar datos **correlacionados** y datos con **variabilidad heterogénea**
- Además, el modelo puede incluir todos los **factores fijos** que se deseen
 - Grupos que diferencian a los pacientes
 - Tratamientos

Modelos lineales mixtos

- Los **modelos de medidas repetidas** pueden ser analizados como modelos lineales mixtos
 - El patrón de la variable respuesta en cada individuo depende de las características de cada individuo, algunas de ellas no observables
 - Las medidas de un individuo están **correlacionadas**
- Las **medidas repetidas** identifican las distintas observaciones realizadas en cada nivel de un factor aleatorio
 - En un análisis longitudinal **los individuos** se consideran como un **factor aleatorio**. Cada individuo es un cluster donde se han tomado distintas medidas
 - Los individuos seleccionados son **una muestra de todos los niveles** posibles del factor aleatorio
 - Los resultados se **extrapolan a la población**, no se tiene especial interés en los individuos seleccionados

Modelo lineal mixto para analizar medidas repetidas

- **El modelo lineal mixto para analizar un factor de medidas repetidas** se presenta de la siguiente forma:
 - Supongamos que tenemos **n individuos (factor aleatorio)** a los cuales se les ha medido **q medidas repetidas**
 - Sea y_{ij} el valor de la variable respuesta del individuo i -ésimo en el periodo j -ésimo (medida repetida)

$$y_{ij} = \mu + \alpha_j + u_i + \varepsilon_{ij} \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, q \end{matrix}$$

- μ es la media global
- α_j es el **efecto de** la j -ésima medida
- u_i es el **efecto aleatorio** del i -ésimo individuo
- ε_{ij} error aleatorio del sujeto i en la medida j

$$u_i \approx N(0, \tau^2)$$

$$\varepsilon_{ij} \approx N(0, \sigma^2)$$

Modelo lineal mixto para analizar medidas repetidas

- **El modelo lineal mixto para analizar un factor de medidas repetidas** que incluye además un **factor fijo** se presenta de la siguiente forma:
 - Supongamos que tenemos **n individuos (factor aleatorio)** a los cuales se les ha medido **q medidas repetidas**, y cada uno pertenece a uno de los **p niveles** de un factor fijo (tratamiento).
 - Sea y_{ijk} el valor de la variable respuesta del individuo i -ésimo en el periodo j -ésimo (medida repetida), que pertenece al grupo k del factor fijo

$$y_{ijk} = \mu + \alpha_j + \beta_k + u_i + \varepsilon_{ij} \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, q \\ k = 1, \dots, p \end{matrix}$$

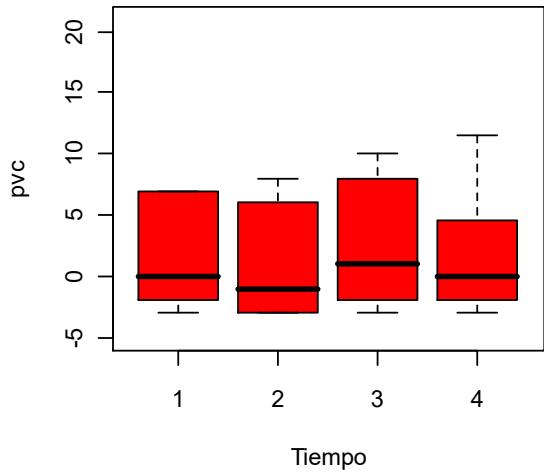
- μ es la media global
- α_j es el **efecto de** la j -ésima medida
- β_k es el **efecto** del k -ésimo grupo
- u_i es el **efecto aleatorio** del i -ésimo individuo $u_i \approx N(0, \tau^2)$
- ε_{ij} error aleatorio del sujeto i en la medida j $\varepsilon_{ij} \approx N(0, \sigma^2)$

Ejemplo: Análisis descriptivo

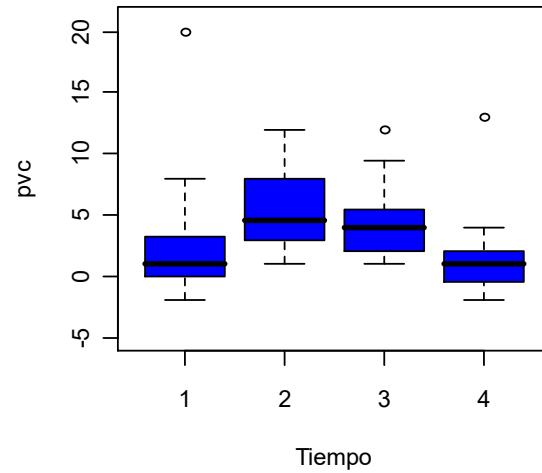
```
> library(nlme)
> ## Fichero Datos: Evolución después de cirugía
> xx <- read.csv(file="C:/Bioestadística con R/Datos/evolución cirugía.csv", header=T)
> dim(xx)
[1] 204   5
> head(xx)
  id tiempo grupo sato2 pvc
1  1      1     1    95   0
2  2      1     1    95   7
3  6      1     1    99  -3
4  3      1     1    99   7
5  4      1     1    93  -2
6  5      1     1    94   0
> xx$tiempo <- factor(xx$tiempo)
> xx$grupo <- factor(xx$grupo)
>
> ## Boxplots de cada Grupo
> pal.col <- c("red", "blue", "green", "yellow" ) ## paleta de colores usados
>
> dev.new()
> par (mfrow=c(2,2))
> for ( i in 1:length(table(xx$grupo)))
+ {
+   boxplot( pvc[grupo == i] ~ tiempo[grupo == i] , data=xx,
+           col=pal.col[i], xlab="Tiempo", ylab="pvc",
+           ylim=c(min(xx$pvc,na.rm=T), max(xx$pvc,na.rm=T)),
+           main=paste("Evolución del PVC en el Grupo =", i ))
+ }
```

Ejemplo: Análisis descriptivo

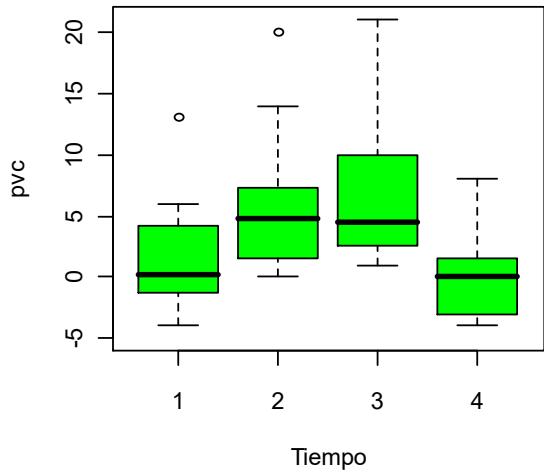
Evolución del PVC en el Grupo = 1



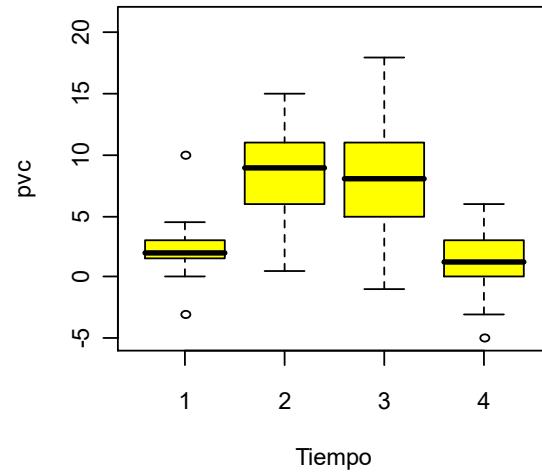
Evolución del PVC en el Grupo = 2



Evolución del PVC en el Grupo = 3



Evolución del PVC en el Grupo = 4



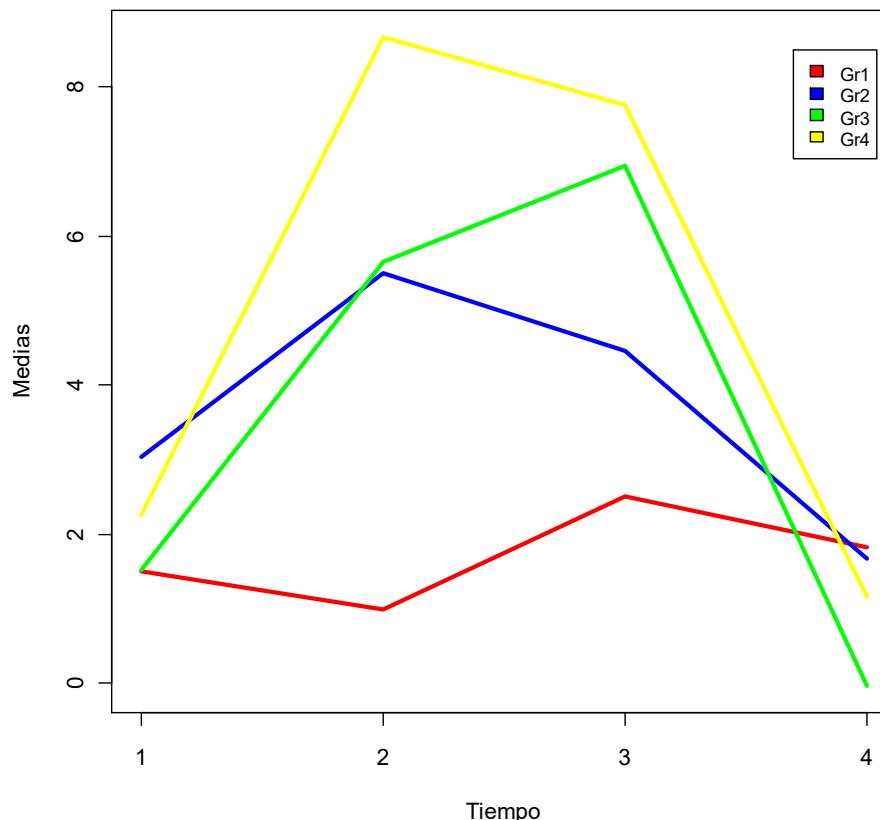
Ejemplo: Análisis descriptivo

```
> ## Cálculo de medias y SD de cada Grupo en cada instante del Tiempo
>
> medias.gr.tm <- tapply ( xx$pvc , list ( xx$grupo, xx$tiempo ) , mean , na.rm=T )
> sd.gr.tm     <- tapply ( xx$pvc , list ( xx$grupo, xx$tiempo ) , sd , na.rm=T )
>
> medias.gr.tm
      1         2         3         4
1 1.500000 1.000000 2.500000 1.8333333
2 3.041667 5.500000 4.454545 1.6666667
3 1.531250 5.656250 6.937500 -0.03571429
4 2.264706 8.676471 7.764706 1.15625000
> sd.gr.tm
      1         2         3         4
1 4.415880 4.816638 5.319774 5.428321
2 5.921679 3.661843 3.538747 3.967672
3 4.306850 5.433595 5.767365 3.554327
4 2.556350 4.103890 5.006613 2.826769
```

- Cada fila de las matrices que contienen las medias y las desviaciones estándar representa a un grupo, y cada columna una medida en el tiempo
- Se utiliza la función *apply()* con *list()* para indicar que se van usar varios factores

Ejemplo: Análisis descriptivo

```
> ## Gráfico de Medias
> dev.new()
> plot( names(table(xx$tiempo)), medias.gr.tm[1, ], type="l", lwd=3,
+       ylim = c( min(medias.gr.tm - sd.gr.tm), max(medias.gr.tm + sd.gr.tm)),
+       col=pal.col[1] , xlab="Tiempo", ylab="Medias", xaxp = c(1,4,3) )
> for ( i in 2:4 )
+ { lines( names(table(xx$tiempo)), medias.gr.tm[i, ], lwd=3, col=pal.col[i] ) }
> legend(3.7,13, c("Gr1","Gr2","Gr3","Gr4"), pal.col , cex = 0.8)
```



Ejemplo: Análisis de medidas repetidas

```
> ## Modelo con efectos principales
> lme.main <- lme( fixed = pvc ~ tiempo + grupo , random = ~1 | id
+                   , data=xx, na.action=na.omit)
> anova(lme.main)
      numDF denDF  F-value p-value
(Intercept)     1    146 70.29743 <.0001
factor(tiempo)   3    146 27.10667 <.0001
factor(grupo)    3     47  1.52235  0.221
```

- La función ***lme()*** de la librería ***nlme*** ajusta un modelo lineal mixto:
 - factores fijos: el tiempo y los grupos
 - factor aleatorio: los individuos (**random = ~1 | id**)
- Hay diferencias significativas en la evolución de la variable “pvc” en el tiempo (P<0.0001)
- No hay diferencias significativas en los grupos (P=0.221)

Ejemplo: Análisis de medidas repetidas

```
> summary(lme.main)
Linear mixed-effects model fit by REML

Random effects:
 Formula: ~1 | id
            (Intercept) Residual
StdDev:     2.732381 3.611140

Fixed effects: pvc ~ factor(tiempo) + factor(grupo)
                Value Std.Error DF   t-value p-value
(Intercept)    0.013645 1.4074883 146  0.009695 0.9923
factor(tiempo)2 3.950980 0.7151122 146  5.524980 0.0000
factor(tiempo)3 3.998217 0.7194544 146  5.557291 0.0000
factor(tiempo)4 -1.170446 0.7285734 146 -1.606490 0.1103
factor(grupo)2  1.977273 1.6396734  47  1.205894 0.2339
factor(grupo)3  1.833015 1.5703932  47  1.167233 0.2490
factor(grupo)4  3.220504 1.5563107  47  2.069320 0.0440
```

- Las diferencias en el tiempo se encuentran en los momentos 2 y 3 respecto al momento basal ($P<0.0001$). Entre la medida final y la medida basal no existen diferencias significativas ($P=0.1103$)
- Aunque no hay diferencias significativas entre grupos, hay una ligera diferencia entre los grupos 1 y 4 ($P=0.044$)

Ejemplo: Análisis de medidas repetidas

```
> #####
> ## Modelo con Interacción
> lme.int <- lme( fixed = pvc ~ tiempo * grupo , random = ~1 | id
+                 , data=xx, na.action=na.omit)
> anova(lme.int)
      numDF denDF  F-value p-value
(Intercept)       1     137 69.60895 <.0001
factor(tiempo)    3     137 30.19064 <.0001
factor(grupo)     3      47  1.50401 0.2257
factor(tiempo) : factor(grupo)  9     137  2.78759 0.0050
```

- Hay diferencias significativas en la evolución de la variable “pvc” en el tiempo ($P<0.0001$)
- No hay diferencias significativas entre los grupos ($P=0.2257$)
- La interacción entre el tiempo y los grupos es significativa ($P=0.0050$). Esto significa que aunque no hay diferencias entre los grupos, la evolución en algunos momentos del tiempo de alguno de los grupos es diferente al resto

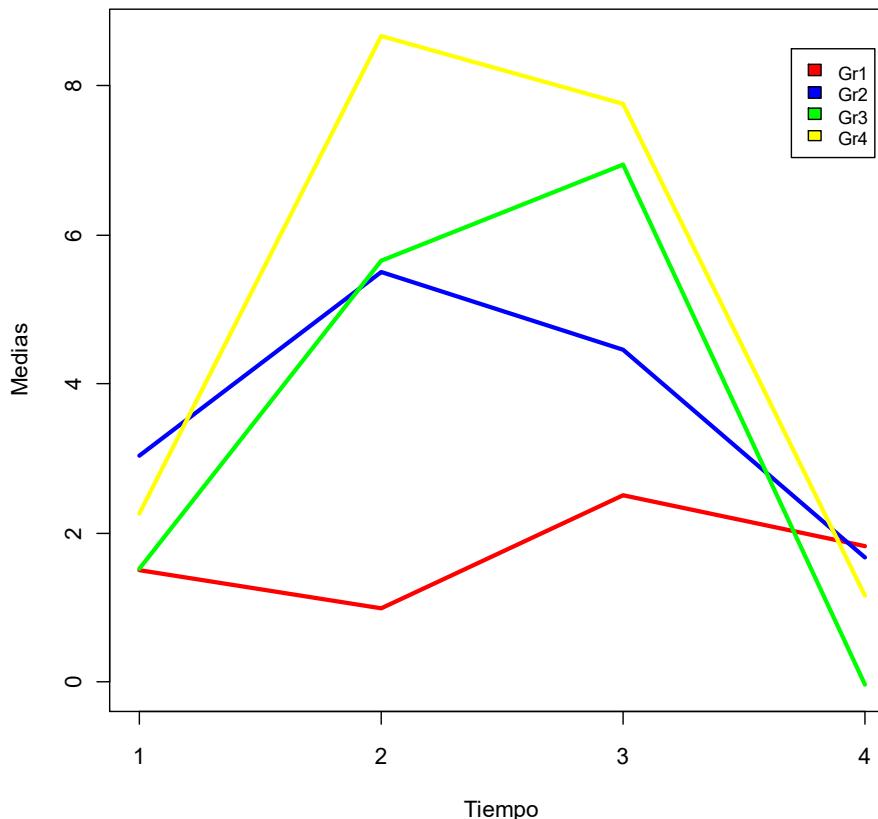
Ejemplo: Análisis de medidas repetidas

```
> summary(lme.int)
```

Fixed effects: pvc ~ factor(tiempo) * factor(grupo)					
	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.500000	1.808385	137	0.8294695	0.4083
factor(tiempo)2	-0.500000	1.976066	137	-0.2530280	0.8006
factor(tiempo)3	1.000000	1.976066	137	0.5060559	0.6136
factor(tiempo)4	0.333333	1.976066	137	0.1686853	0.8663
factor(grupo)2	1.541667	2.214810	47	0.6960717	0.4898
factor(grupo)3	0.031250	2.120519	47	0.0147370	0.9883
factor(grupo)4	0.764706	2.103441	47	0.3635500	0.7178
factor(tiempo)2:factor(grupo)2	2.958333	2.420177	137	1.2223624	0.2237
factor(tiempo)3:factor(grupo)2	0.327512	2.442500	137	0.1340887	0.8935
factor(tiempo)4:factor(grupo)2	-1.708333	2.420177	137	-0.7058713	0.4815
factor(tiempo)2:factor(grupo)3	4.625000	2.317143	137	1.9959925	0.0479
factor(tiempo)3:factor(grupo)3	4.406250	2.317143	137	1.9015874	0.0593
factor(tiempo)4:factor(grupo)3	-1.936159	2.344586	137	-0.8257999	0.4104
factor(tiempo)2:factor(grupo)4	6.911765	2.298481	137	3.0071009	0.0031
factor(tiempo)3:factor(grupo)4	4.500000	2.298481	137	1.9578146	0.0523
factor(tiempo)4:factor(grupo)4	-1.681745	2.309912	137	-0.7280557	0.4678

- Al igual que en los modelos de regresión, si una interacción es significativa, puede que los efectos principales dejen de ser significativos

Ejemplo: Análisis de medidas repetidas



- La significación de la interacción viene determinada porque los grupos se comportan de forma diferente en el momento 2:
 - El grupo 4 tiene una evolución diferente en el momento 2 ($P=0.0031$)
 - El grupo 3 también presenta ligeras diferencias en el momento 2 ($P=0.0479$)

Test de Friedman de medidas repetidas

- El test de Friedman es una **prueba no parámetrica para el análisis de medidas repetidas**
 - Alternativa a los modelos lineales, que requieren normalidad y homocedasticidad
 - Basado en **rangos**, ordena los datos por filas o bloques (individuos), reemplazándolos por sus respectivo orden
 - Es una extensión del test de Wilcoxon para datos pareados
- El test de Friedman no permite incorporar otros factores al análisis, solo se evalúa el factor de medidas repetidas

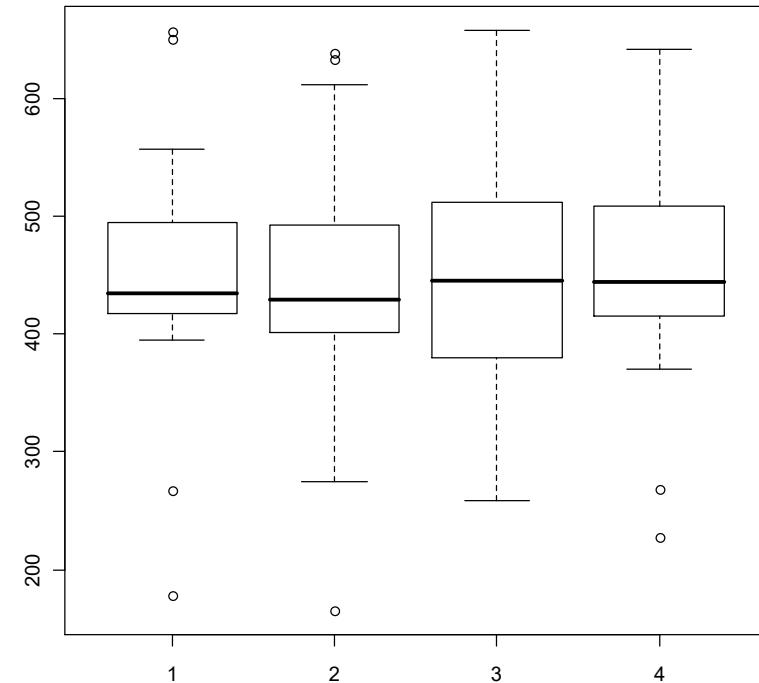
Ejemplo: Test de Friedman

```
> yy <- read.csv(file="C:/Bioestadistica con R/Datos/peak.csv", header=T)
>
> ## Preparar los datos y descriptivo
> id <- rep ( yy$subject, 4)
> medida <- c ( rep(1,nrow(yy)), rep (2,nrow(yy)), rep (3,nrow(yy)), rep (4,nrow(yy)) )
> peak <- c ( yy$wpfm_1, yy$wpfm_2, yy$mwm_1, yy$mwm_2 )
> ## Boxplot
> dev.new()
> boxplot( peak ~ medida )
>
> ## Normalidad
> shapiro.test ( yy$wpfm_1 )
W = 0.9199, p-value = 0.1469
> shapiro.test ( yy$wpfm_2 )
W = 0.932, p-value = 0.2348
> shapiro.test ( yy$mwm_1 )
W = 0.966, p-value = 0.7458
> shapiro.test ( yy$mwm_2 )
W = 0.9575, p-value = 0.5849

> ## Test de igualdad de varianza
> bartlett.test ( peak ~ medida )

Bartlett test of homogeneity of variances

data: peak by medida
Bartlett's K-squared = 0.094, df = 3, p-value = 0.9925
```



Ejemplo: Test de Friedman

```
> ## Test basado en modelos lineales mixtos
> anova( lme( fixed = peak ~ medida , random = ~1 | id ) )
      numDF denDF   F-value p-value
(Intercept)     1     50 269.65606 <.0001
medida         1     50   0.72391  0.3989
>
> ## Test de Friedman
> friedman.test(peak, medida, id)

  Friedman rank sum test
data: peak, medida and id
Friedman chi-squared = 1.9464, df = 3, p-value = 0.5836

> ## Test de Friedman. Alternativa para ejecutarlo
> all.peak <- cbind ( yy$wpfm_1, yy$wpfm_2, yy$mwm_1, yy$mwm_2 )
> friedman.test(all.peak)

Friedman chi-squared = 1.9464, df = 3, p-value = 0.5836
```

- Se cumplen los supuestos de normalidad y homocedasticidad, y por tanto se podría aplicar en análisis basado en los modelos lineales mixtos
- El test basado en modelos lineales mixtos ($P=0.3989$) y el test de Friedman ($P=0.5836$) nos indican que no hay diferencias significativas entre las 4 medidas repetidas

Ejercicio

- Fichero de datos: evolución cirugía
- Realizar un **análisis de medidas repetidas** con la variable “sato2”, para ver si hay diferencias entre grupos y en el tiempo
 - Ajustar el modelo incluyendo **la interacción** entre grupo y tiempo

Estadística Aplicada a la Investigación Biomédica con R

22 Comparaciones Múltiples

- ✓ Problema de comparaciones múltiples
- ✓ Corrección por Bonferroni
- ✓ False Discovery Rate
- ✓ Método de Benjamini-Hochberg
- ✓ q-value

Problema de Comparaciones Múltiples

- **Error de tipo I** es rechazar H_0 cuando es cierta. **Falso positivo**
- **Error de tipo II** es aceptar H_0 cuando es falsa. Falso negativo
- Se suele tomar un **nivel de significación α** , para controlar el error de tipo I, que son los falsos positivos. Habitualmente $\alpha = 0.05$
- Llamamos **p-valor** de un test a la probabilidad de obtener, suponiendo que H_0 sea cierta, un resultado al menos tan extremo como el que realmente se ha obtenido en la muestra observada. El test se rechaza cuando p-valor < α
- El problema de **comparaciones múltiples** surge cuando se realizan varios tests simultáneamente, que tiene como consecuencia el incremento en el error de tipo I
 - Si se considera un nivel de significación $\alpha = 0.05$, este valor es válido solamente si se realiza un único test. Se está controlando la probabilidad de un falso positivo
 - Si se hacen 100 tests, se espera que en promedio haya un 5% de falsos positivos, aunque todas las hipótesis fueran falsas

Problema de Comparaciones Múltiples

- Cuando se realizan m contrates de hipótesis, tenemos la siguiente tabla con los **resultados** que se han obtenido

	Tests significativos	Tests no significativos	Total Tests
H_0 ciertas	F	$m_0 - F$	m_0
H_0 falsas H_1 ciertas	T	$m_1 - F$	m_1
Total	S	$m - S$	m

- m es el número de hipótesis contrastadas, de las que m_0 son ciertas y m_1 falsas, pero estas dos cantidades son desconocidas
- S es el número de tests considerados **significativos**, independientemente de que las hipótesis sean ciertas o falsas, y es una cantidad conocida
- De los S tests significativos, T han sido correctamente declarados significativos, y habrá F hipótesis que eran ciertas, que son llamados **falsos descubrimientos**

Problema de Comparaciones Múltiples

- Se puede estimar la **tasa de falsos positivos**, false positive rate, como la probabilidad de rechazar la hipótesis nula (test significativo) siendo cierta:
 - El cociente del FPR se calcula sobre el total de hipótesis ciertas, que es desconocido

$$FPR = \frac{F}{m_0}$$

- Se puede estimar la **tasa de falsos descubrimientos**, false discovery rate, como la probabilidad de que la hipótesis nula sea cierta, a pesar de haber sido rechazada (test significativo)
 - El cociente del FDR se calcula sobre el total de tests significativos

$$FDR = \frac{F}{S}$$

Corrección de Bonferroni

- La **corrección de Bonferroni** se usa para asegurar que la probabilidad de obtener **al menos un falso positivo** entre todas las comparaciones sigue estando por debajo del nivel de significación $\alpha = 0.05$
 - Contrastá la “hipótesis nula general” de que todas las hipótesis nulas testadas son ciertas, de forma simultánea, lo que no suele ser de interés
- La **corrección de Bonferroni** consiste en tomar un nivel de significación corregido = α / m , donde m es el número de tests
- Equivalentemente, se pueden calcular un **p-valor ajustado** = $m \times p\text{-valor}$ para cada uno de los tests, y aplicar el nivel de significación habitual α
- Se considera un **método conservador**, sobre todo, cuando el número de comparaciones es elevado
- Otro problema que presenta es que los p-valores ajustados dependen del número de tests, lo que es una idea poco intuitiva

Ejemplo: Comparaciones múltiples. Datos simulados

```
> ## Número de observaciones y número de variables
> n.ind = 80
> num.var = 10000
> ## Definición de la variable respuesta binaria
> var.gr = c ( rep ("A" , n.ind/2 ) , rep ("B" , n.ind/2 ) )
> var.gr
[1] "A" "A"
[21] "A" "A"
[41] "B" "B"
[61] "B" "B"
> ## Definición y análisis de una variable independiente
> var.x = rnorm ( n.ind , mean = 5 , sd = 2 )
> ## Medias, SD y t de Student para los 2 grupos
> tapply ( var.x , var.gr, mean )
      A          B
4.709394 5.015168
> tapply ( var.x , var.gr, sd )
      A          B
2.140059 1.946776
> t.test ( var.x ~ var.gr )
t = 2.4761, df = 77.311, p-value = 0.5108
```

- Se genera **una variable binaria** definiendo 2 grupos “A” y “B” (n=80)
- Se crea una variable independiente, asignando **aleatoriamente** valores de una distribución normal. Se realiza un test **t de Student** para estudiar la diferencia de esa variable en los grupos. El p-valor = 0.51 indica que no hay diferencias significativas

Ejemplo: Comparaciones múltiples. Datos simulados

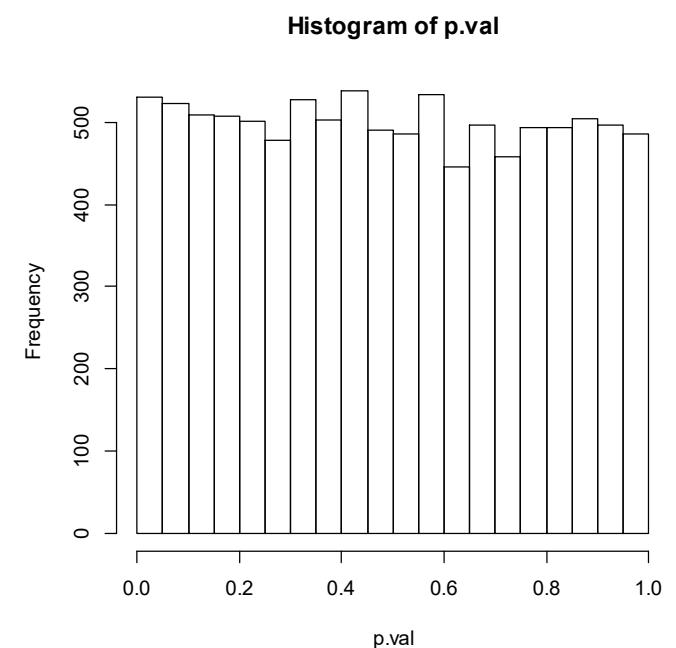
```
> ## Generamos una matriz con muchas variables con distribuciones normales
> yy = matrix ( NA, n.ind , num.var )
> yy = apply ( yy , 2 ,
+               function ( w.col ) { w.col = rnorm ( n.ind , mean = 5 , sd = 2 ) } )
> dim (yy)
[1] 80 10000
> yy [1:5 , 1:5 ]
      [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 4.6551001 4.223183 4.058378 1.440447 4.156688
[2,] 3.2760800 7.839703 8.914302 4.818339 4.165585
[3,] 3.5391370 4.997973 4.925947 5.307875 4.124802
[4,] 0.2863208 4.043007 5.528752 6.084604 6.861643
[5,] 7.4214537 4.222200 5.542918 7.419333 6.174869
>
> ## Test t de Student para todas las variables
> p.val = apply ( yy , 2 , function ( w.col ) { t.test ( w.col ~ var.gr )$p.value } )
> length(p.val)
[1] 10000
> head(p.val)
[1] 0.81267592 0.01084827 0.77155845 0.06978888 0.27783414 0.72733104
```

- Se crean ahora **10000 variables** independientes con el mismo proceso, es decir, se asignan **aleatoriamente** valores de una distribución normal $N(5,2)$ a cada grupo
- Se realizan 10000 test **t de Student** para estudiar las diferencias de cada una de las variables en los 2 grupos, y se guardan los **p-valores** obtenidos

Ejemplo: Comparaciones múltiples. Datos simulados

```
> sort(p.val) [1:6]
[1] 0.0001780096 0.0002877687 0.0003108818 0.0004006488 0.0004923371 0.0006667453
> sum ( p.val < 0.05 )
[1] 493
> dev.new(); hist ( p.val , breaks = 20 , main="" )
>
> ## Bonferroni. Nivel de significación es alfa/p
> 0.05 / num.var
[1] 5e-06
> sum ( p.val < 0.05 / num.var )
[1] 0
```

- Se han obtenido con datos aleatorios p-valores muy pequeños, muy por debajo de 0.05
- Si se toma $\alpha = 0.05$ hay 493 p-valores < 0.05 , y por tanto, 493 variables se considerarían significativas. Aproximadamente **un 5%**
- Los p-valores generados con datos simulados siguen una **distribución uniforme**
- Al aplicar **la corrección de Bonferroni**, ninguno de los tests sería significativo



False Discovery Rate (FDR)

- Cuando se realiza una gran cantidad de tests, no es útil controlar la probabilidad de cometer un error de tipo I
 - Sobre todo en las situaciones en las que se espera que haya muchos tests significativos, como suele suceder en algunos **análisis exploratorios**
 - Se necesita potencia para encontrar asociaciones, y a su vez es aceptable cometer algunos errores
- La alternativa que se suele usar es controlar la **tasa de falsos descubrimientos**, false discovery rate (FDR)
- El **false discovery rate** se define como la proporción esperada de hipótesis nulas que son verdad, entre todas las hipótesis declaradas significativas
 - Alternativamente, el FDR es la proporción de tests significativos que no lo son

False Discovery Rate (FDR)

- Para establecer un **control del FDR** para un conjunto de tests, se define un límite, de tal forma, que entre todos los tests significativos, la proporción de falsos positivos no supera ese límite
- Es una idea bastante intuitiva: si se establece un control del FDR al 10%, significa que menos del 10% de los resultados dados por significativos son falsos positivos

Control del FDR. Método Benjamini & Hochberg

- El **método de Benjamini y Hochberg (BH)** para controlar el **FDR** a un porcentaje α en un estudio donde se han realizado m tests consiste en:
 - Se ordenan los p-valores de los m tests $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(m)}$
 - Se define k como la última posición para la que se cumple $p_{(k)} \leq \alpha \cdot \frac{k}{m}$
 - Se consideran significativos todos los tests hasta esa posición k
 - Se observa que cada test tiene diferente criterio para ser rechazado (se llaman métodos step-down)
- Los **p-valores ajustados** por el método de **Benjamini-Hochberg** se obtienen:

$$p.\text{adj}_{(j)} = \min_{k=j, \dots, m} \left\{ \frac{m}{k} \cdot pval_{(k)} \right\}$$

Ejemplo: Comparaciones múltiples

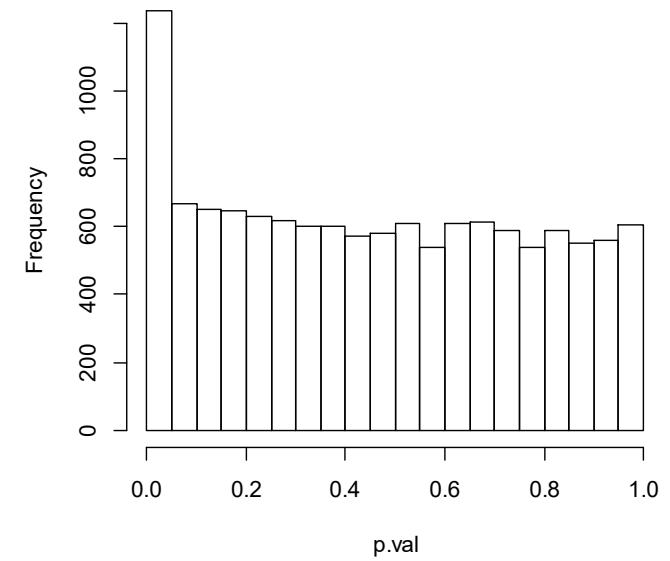
```
> ## Fichero ALL: 12625 variables genéticas y una que define 2 clases (última col)
> xx <- read.delim( "C://Bioestadistica con R/Datos/ALL.txt", sep=" ")
> dim(xx)
[1] 79 12626
> ## Número de variables, número de tests
> n.var = ncol(xx) - 1
> head(names(xx))
[1] "X1000_at"    "X1001_at"    "X1002_f_at"  "X1003_s_at"  "X1004_at"    "X1005_at"
> tail(names(xx))
[1] "AFFX.TrpnX.M_at"      "AFFX.YEL002c.WBP1_at" "AFFX.YEL018w._at"
[4] "AFFX.YEL021w.URA3_at" "AFFX.YEL024w.RIP1_at"   "mol.biol"
> table( xx$mol.biol )
BCR/ABL     NEG
      37      42
> ## t de Student para cada una de las variables
> p.val = apply ( xx[ , - ncol(xx) ] , 2 ,
+                 function ( w.col ) { t.test ( w.col ~ xx$mol.biol )$p.value } )
> length(p.val)
[1] 12625
> sum ( p.val < 0.05 )
[1] 1237
```

- Se realizan 12625 tests **t de Student** para cada una de las variables genéticas, para analizar cuáles son diferentes en los dos grupos de tumores
- Se salvan los **p-valores** obtenidos. Hay 1237 variables con un p-valor < 0.05, que supone casi un 10% del total de variables, muy superior al 5% esperado por azar

Ejemplo: Comparaciones múltiples. Bonferroni

```
> dev.new(); hist ( p.val , breaks = 20 , main="" )
> ## Bonferroni. Nivel de significación es alfa/p
> sum ( p.val < 0.05 / n.var )
[1] 20
> w.p.adj.bonf = p.val * n.var                                ## p-valores ajustados
> w.p.adj.bonf [ p.adj.bonf > 1 ] = 1
> sum ( w.p.adj.bonf < 0.05 )
[1] 20
> ## Bonferroni
> p.adj.bonf = p.adjust ( p.val , method = "bonferroni" )
> sum ( p.adj.bonf == w.p.adj.bonf )                            ## Chequeamos que coinciden
[1] 12625
> sum ( p.adj.bonf < 0.05 )
[1] 20
> var.sign.bonf = names(xx) [ p.adj.bonf < 0.05 ] ## Variables significativas
> length(var.sign.bonf)
[1] 20
```

- La **distribución** de los p-valores **no es uniforme**, hay muchos p-valores bajos
- Se usa la función ***p.adjust()*** para obtener los p-valores ajustados por la corrección de Bonferroni, aunque se puede hacer manualmente
- Aplicando la **corrección de Bonferroni**, hay 20 variables significativas



Ejemplo: Control del FDR con el método BH

```
> ## Benjamini-Hochberg
> p.adj.BH = p.adjust( p.val , method = "BH" )
> sum( p.adj.BH < 0.05 )
[1] 163
> ## Seleccionar las variables significativas por BH al FDR del 5%
> var.sign.BH = names(xx) [ p.adj.BH < 0.05 ]
> ## Explicando BH
> m = length(p.val)
> d = 0.05
> i = 1:m
> p.val.sort = sort(p.val)
> p.thr = 0.05 * ( i / m )
> head(p.val.sort)
  x1636_g_at    x39730_at    x1635_at    x1674_at    x40202_at    x40504_at
[1] 1.792370e-13 1.206402e-12 7.102753e-10 6.105794e-09 1.797179e-08 2.272148e-08
> head(p.thr)
[1] 3.960396e-06 7.920792e-06 1.188119e-05 1.584158e-05 1.980198e-05 2.376238e-05
> sum(p.val.sort < p.thr )      ## Contando significativos según BH
[1] 163
```

- Se usa la función ***p.adjust()*** para obtener los p-valores ajustados por la corrección de Benjamini Hochberg (BH)
- Aplicando **el método de Benjamini Hochberg** para controlar el FDR al 5%, hay 163 variables significativas, frente a solo 20 que había por la corrección de Bonferroni
- Se pueden establecer los thresholds individuales a los que son rechazados cada uno de los p-valores, y comprobar que son 163 los que se dan por significativos

Ejemplo: Control del FDR con el método BH

```
> ## p.ajustados por BH obtenidos manualmente
> sort(p.val.sort) [1:5]
  x1636_g_at    x39730_at    x1635_at    x1674_at    x40202_at
1.792370e-13 1.206402e-12 7.102753e-10 6.105794e-09 1.797179e-08
> sort(p.adj.BH) [1:5]
  x1636_g_at    x39730_at    x1635_at    x1674_at    x40202_at
2.262867e-09 7.615411e-09 2.989075e-06 1.927141e-05 4.537877e-05
>
> min ( sort(p.val.sort) [1:m] * ( m / 1:m ) )
[1] 2.262867e-09
> min ( sort(p.val.sort) [2:m] * ( m / 2:m ) )
[1] 7.615411e-09
> min ( sort(p.val.sort) [3:m] * ( m / 3:m ) )
[1] 2.989075e-06
>
> w.p.adj.BH = rep ( NA, m )                      ## p-valores ajustados
> for ( k in 1:m )
+ {
+   pos = which ( names ( sort(p.val.sort) [k] ) == names(xx) )
+   w.p.adj.BH [ pos ] = min ( sort(p.val.sort) [k:m] * ( m / k:m ) )
+ }
> w.p.adj.BH [ w.p.adj.BH > 1 ] = 1
> sum ( p.adj.BH == w.p.adj.BH )                   ## Chequeamos que coinciden
[1] 12625
```

- Para calcular manualmente los p-valores ajustados por **el método de Benjamini Hochberg** hay que hacerlo por el orden de los p-valores, pero sabiendo la posición de la variable dentro de ese orden

q-valor

- Se define el **q-valor** de un test como el mínimo FDR que se puede obtener si se consideran significativos ese test y todos los que tenga menor p-valor
 - El q-valor es la proporción de falsos positivos entre todos los tests iguales o más extremos que el observado
 - Es una extensión del concepto de FDR que en lugar de estar asociado al conjunto de tests significativos, se asocia a cada test de forma individual
 - Si el q-valor de una hipótesis es 0.021 significa que aproximadamente un 2.1% de los tests más significativos que el que se está considerando, son falsos positivos
- Si se consideran como significativos todos aquellos tests cuyo q-valor sea menor que Φ , se está seguro que el $FDR \leq \Phi$
- Hay diferentes métodos para calcular el q-valor, que se basan en la distribución de los p-valores
 - Para controlar el q-valor, se utiliza un parámetro λ que se puede optimizar, y tiene relación con qué parte del histograma de p-valores se puede considerar uniforme
 - Se estima $\pi_0 = m_0 / m$, hipótesis nulas verdaderas entre todas las contrastadas

Ejemplo: Control del FDR con el q-value

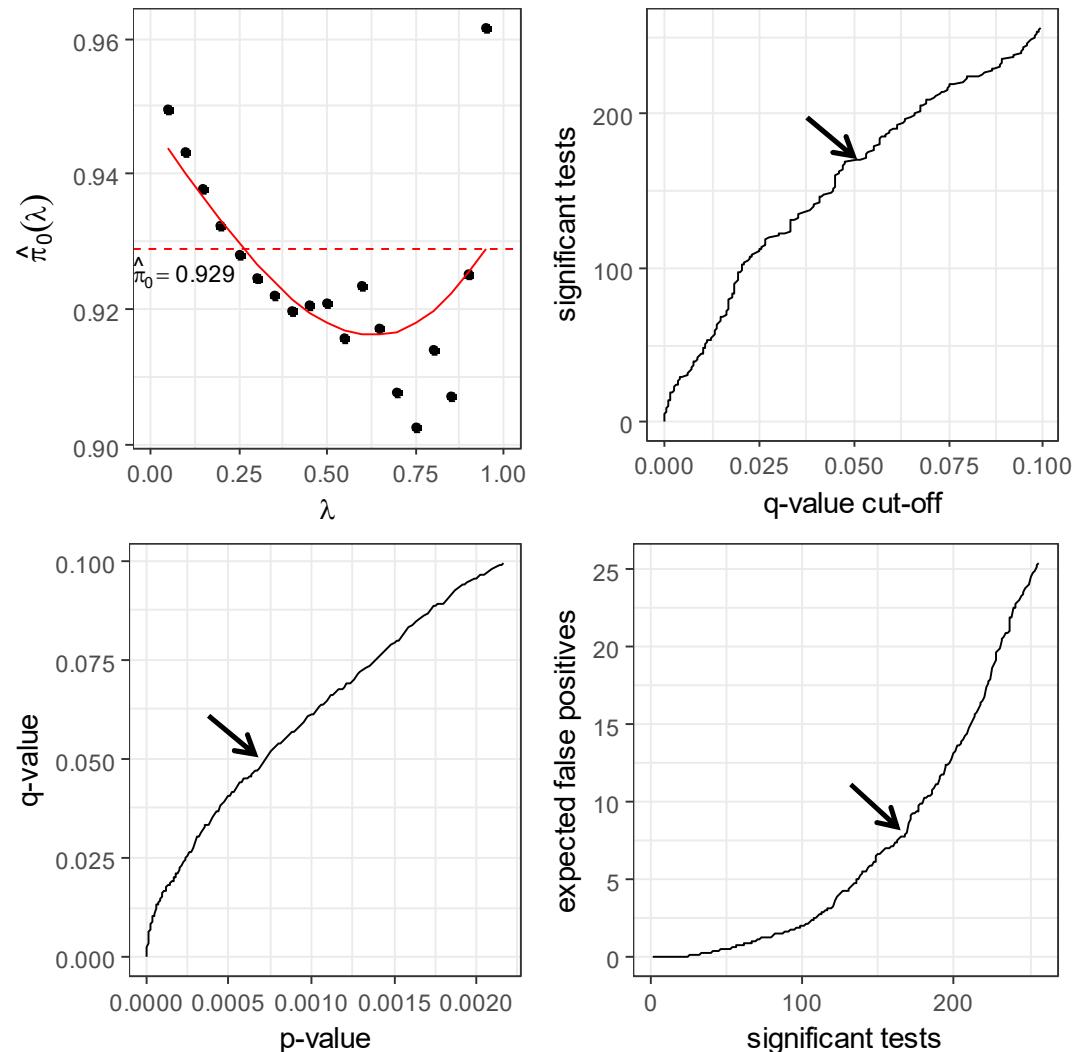
```
> ## q-values
> library(qvalue)
> qval.out = qvalue ( p.val )
> ## Estimación del FDR si se consideran p-values < 0.01 o 0.001
> ## Mayor q-value entre todos los que cumple la condición sobre p-values
> max ( qval.out$qvalues [ qval.out$pvalues <= 0.01 ] )
[1] 0.2249427
> max ( qval.out$qvalues [ qval.out$pvalues <= 0.001 ] )
[1] 0.06123077
> ## Seleccionar las variables significativas por qvalue al FDR del 5%
> qval.out = qvalue ( p.val , fdr.level = 0.05 )
> table ( qval.out$significant )
FALSE  TRUE
12456 169
> max ( p.val [ qval.out$significant == T ] )    ## p-value para significación
[1] 0.0006826484
> var.sign.qval = names(xx) [ qval.out$significant == TRUE ]
> length(var.sign.qval)
[1] 169
```

- Se usa la función ***qvalue()*** de la librería ***qvalue*** (Bioconductor) para usar el método
- Se puede estimar el FDR que tendríamos si se consideran significativos los p-valores menores que un determinado límite. Por ejemplo, el FDR sería 0.225 para un límite de 0.01, que significa que el 22.5% de los p-valores ≤ 0.01 serían falsos positivos
- Aplicando **el método q-value** para controlar el FDR al 5%, hay 169 variables significativas, frente a 163 según el método de Benjamini Hochberg

Ejemplo: Control del FDR con el q-value

```
> ## Plot del q-value  
> dev.new()  
> plot(qval.out)
```

- Evaluación de la estimación del valor π_0 , parámetro del método
- Número de tests significativos respecto a cada límite del q-valor (169 con q-valor=0.05)
- Relación entre q-valor y p-value (significación con p-val=0.00068 para q-valor=0.05)
- Número de falsos positivos esperados en función del límite del q-valor (8 aprox. para 169)



Comparaciones múltiples. Comentarios

- El control de FDR con el **método de Benjamini-Hochberg** es más conservador que mediante el cálculo de los **q-valores**
- Los métodos de **control del FDR** tienen sentido para un número grande de tests, y siempre que se hayan obtenido bastantes p-valores significativos
- Se deben diseñar **estudios de confirmación** que validen los resultados encontrados. En estos estudios se debería ser más exigente
- La **corrección de Bonferroni** se puede usar cuando se realizan pocos tests o hay muy pocos significativos
 - También cuando declarar un test significativo siendo incorrecto es grave

Ejercicio

- Fichero de datos: Virco
 - Variable respuesta: sens.NFV (binaria)
 - Variables predictoras: Todos los predictores binarios [1:89]
- Realizar una comparación con el **test de Fisher** para cada una de las variables predictoras con la variable respuesta
- Aplicar los métodos de **comparaciones múltiples** a los p-valores obtenidos:
 - Método de Bonferroni con $\alpha = 0.05$
 - Control del FDR al 5% con Benjamini-Hochberg
 - Control del FDR al 5% con el método q-value

Estadística Aplicada a la Investigación Biomédica con R

23 Análisis de Componentes Principales (PCA)

- ✓ **Análisis de Componentes Principales**
- ✓ **Interpretación de las Componentes**
- ✓ **Número de Componentes Principales**

Análisis de Componentes Principales (PCA)

- El **Análisis de Componentes Principales** (Pearson 1901, Hotelling 1933) es una **técnica estadística multivariante de reducción de datos** de un conjunto de **variables correlacionadas**
- Permite **reducir la dimensionalidad** del conjunto de datos original
 - Encuentra un número reducido de dimensiones donde los datos pueden ser representados lo “mejor posible”
- El Análisis de Componentes Principales se considera una **técnica no supervisada** porque no hay una variable respuesta que supervise el análisis
- El Análisis de Componentes Principales también permite **visualizaciones** de conjuntos de datos en un espacio de **pocas dimensiones**
 - Permite visualizar las **observaciones** y las **variables**

Análisis de Componentes Principales (PCA)

- El Análisis de Componentes Principales es una **técnica descriptiva** que permite **examinar las correlaciones** existentes en un conjunto de variables
- **Describe la variabilidad** de un conjunto de datos en función de un conjunto de **variables independientes**, no correlacionadas, llamadas **componentes principales (CP)** que son **combinaciones lineales** de las variables originales

$$CP_1 = a_{11} \cdot X_1 + a_{12} \cdot X_2 + \dots + a_{1p} \cdot X_p$$

$$CP_2 = a_{21} \cdot X_1 + a_{22} \cdot X_2 + \dots + a_{2p} \cdot X_p$$

.....

$$CP_p = a_{p1} \cdot X_1 + a_{p2} \cdot X_2 + \dots + a_{pp} \cdot X_p$$

- El **número total** de componentes principales es: $\min(N, p)$

Análisis de Componentes Principales (PCA)

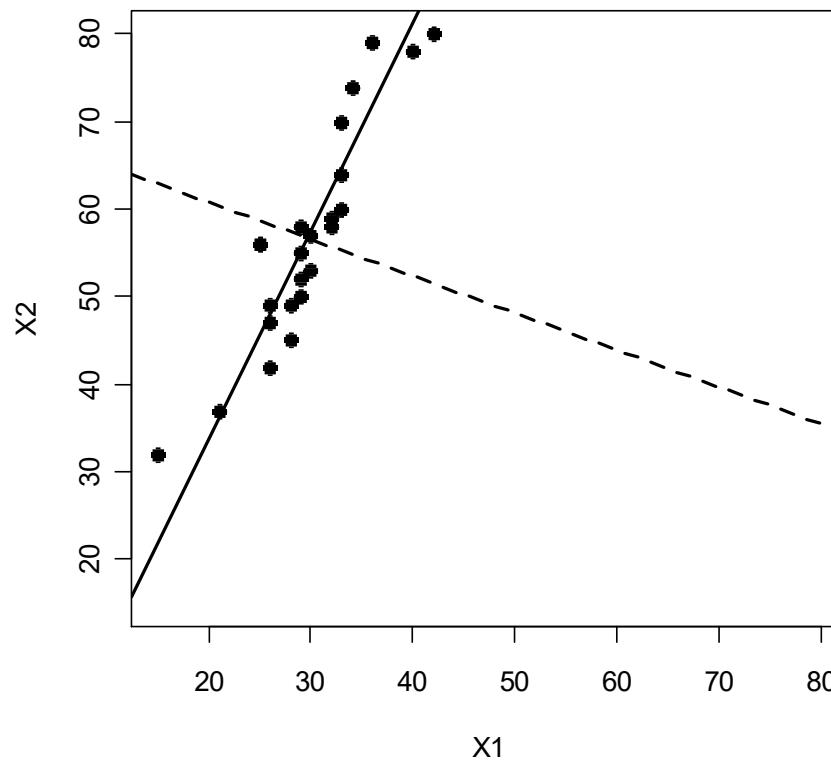
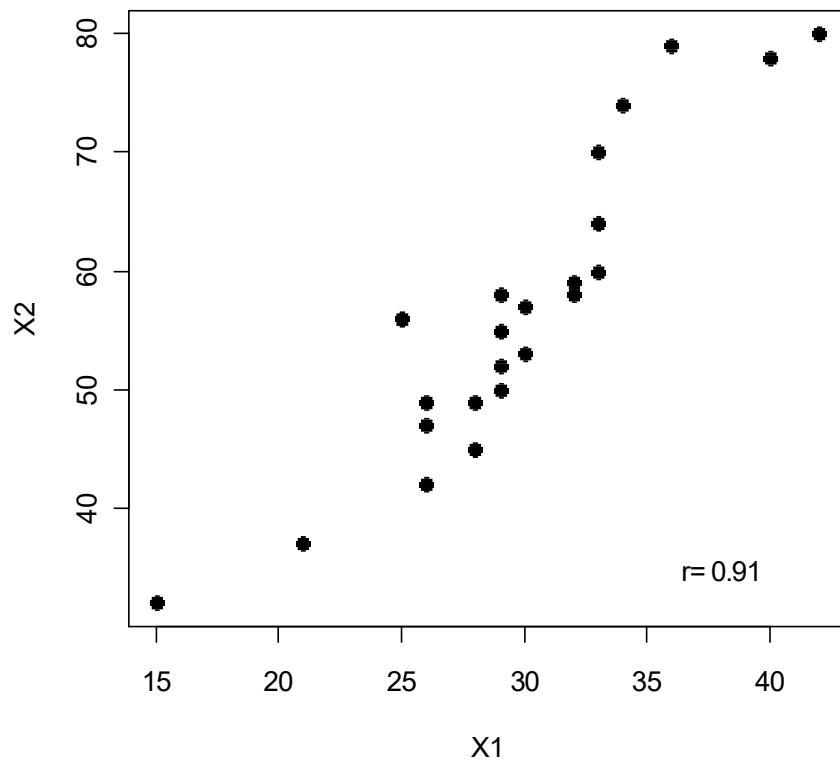
- Las **componentes principales** (CPs) son **independientes** entre sí y se van extrayendo de forma secuencial explicando la **máxima variabilidad** posible
 - La 1^a CP es la combinación lineal de las variables originales que explica la **máxima variabilidad**
 - La 2^a CP es la combinación lineal de las variables originales que explica el **máximo de la variabilidad restante**, y es **independiente** de la 1^o CP
 - La 3^a CP ...

$$\text{Var}(\text{CP}_1) \geq \text{Var}(\text{CP}_2) \ge; \dots \geq \text{Var}(\text{CP}_p)$$

$$\text{Corr}(\text{CP}_i, \text{CP}_j) = 0 \quad \forall i \neq j$$

- Las M primeras componentes, que explican una cantidad sustancial de la variabilidad original, proporcionan **una descripción más simple de los datos**. Proporcionan la mejor aproximación en M dimensiones en términos de **distancia euclídea**
 - Las últimas componentes explican una varianza muy pequeña, y podrían ser descartadas

Análisis de Componentes Principales (PCA)



- La 1º CP está en la dirección de la **máxima variabilidad**. No es una recta de regresión, es la línea que está **más próxima** a todas las **observaciones**, usando la **distancia euclídea**
- La 2º CP es perpendicular a la 1º CP, y explica la poca “**variabilidad restante**”
- Cada punto se podría “simplificar” proporcionando su valor en la 1º CP (proyección del punto en la recta), con una **pérdida mínima de información**

Análisis de Componentes Principales (PCA)

- Las componentes principales se extraen a partir de **S, la matriz de covarianzas** entre todas las variables originales
- Si las variables están medidas en diferentes unidades, las CPs se deberían extraer a partir de **R, la matriz de correlaciones lineales de Pearson**
 - La matriz de correlaciones R es la matriz de covarianzas de las **variables estandarizadas**, restando la media y dividiendo por la SD

$$X \longrightarrow R \longrightarrow \text{CPs}$$

- Las **varianzas explicadas** por cada CP son los **valores propios λ_i** de S o de R (eigenvalues). Si las CPs se han extraído de R, se cumple:

$$\sum_{i=1}^p \lambda_i = \sum_{i=1}^p \text{Var}(\text{CP}_i) = p$$

Interpretación de las Componentes Principales

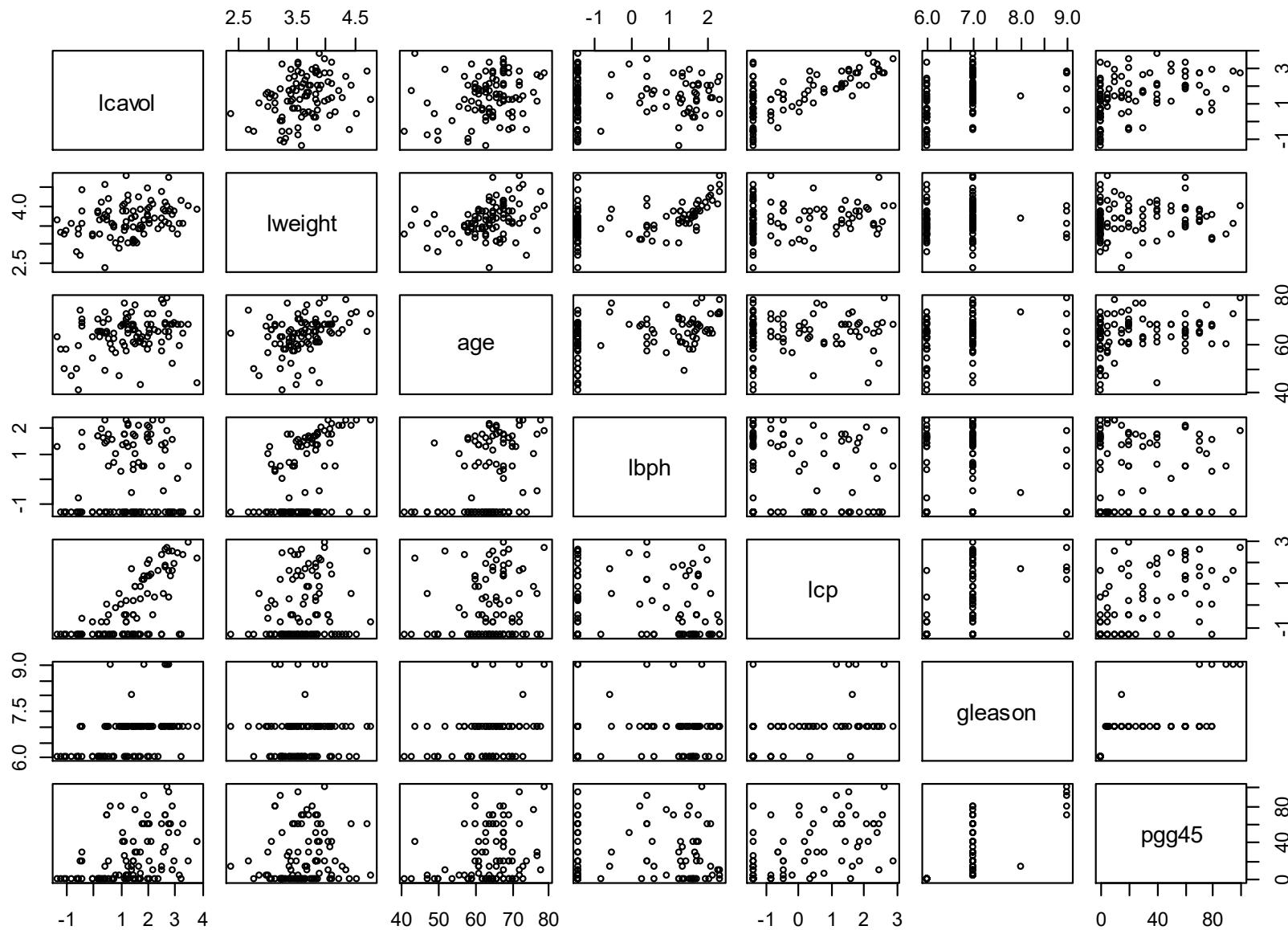
- El Análisis de Componentes Principales proporciona una descripción más simple de los datos, y se debe intentar **interpretar las componentes principales más importantes**
- Los coeficientes a_{ij} son llamados **cargas factoriales** (loadings) y miden la **correlación** entre la componente principal CP_i y la variable X_j
 - Los coeficientes a_{ij} que definen las CPs son los **vectores propios** de S o de R (eigenvectors)
- Cada **componente principal se interpretará** en función de las **variables** que tengan **cargas factoriales altas** en esa componente
- Variables que **no están correlacionadas** con ninguna otra pueden tener “su” componente principal
 - El ACP solo tiene sentido si hay variables que están **correlacionadas**

Ejemplo: Análisis de Componentes Principales

```
> xx <- read.delim("C://Data Mining con R/Datos/Prostate.txt", sep="\t", header=T)
> dim(xx)
[1] 97 11
> head(xx)
  ID lcavol lweight age    lbph svi      lcp gleason pgg45      lpsa train
1  1 -0.5798185 2.769459 50 -1.386294 0 -1.386294       6 0 -0.4307829 TRUE
2  2 -0.9942523 3.319626 58 -1.386294 0 -1.386294       6 0 -0.1625189 TRUE
3  3 -0.5108256 2.691243 74 -1.386294 0 -1.386294       7 20 -0.1625189 TRUE
> ## Variable binaria con la mediana de lpsa (resp.) como pto de corte (para gráficos)
> lpsa.gr = as.numeric ( xx$lpsa > median(xx$lpsa) )
>
> ## Correlaciones de las variables predictoras continuas
> cor.matr = cor(xx[c(2:5,7:9)], method = c("pearson") )
> round( cor.matr, 4 )
      lcavol lweight    age    lbph      lcp gleason    pgg45
lcavol  1.0000  0.2805  0.2250  0.0273  0.6753  0.4324  0.4337
lweight  0.2805  1.0000  0.3480  0.4423  0.1645  0.0569  0.1074
age     0.2250  0.3480  1.0000  0.3502  0.1277  0.2689  0.2761
lbph    0.0273  0.4423  0.3502  1.0000 -0.0070  0.0778  0.0785
lcp     0.6753  0.1645  0.1277 -0.0070  1.0000  0.5148  0.6315
gleason 0.4324  0.0569  0.2689  0.0778  0.5148  1.0000  0.7519
pgg45   0.4337  0.1074  0.2761  0.0785  0.6315  0.7519  1.0000
> dev.new()
> pairs(xx[c(2:5,7:9)], cex=0.8)
```

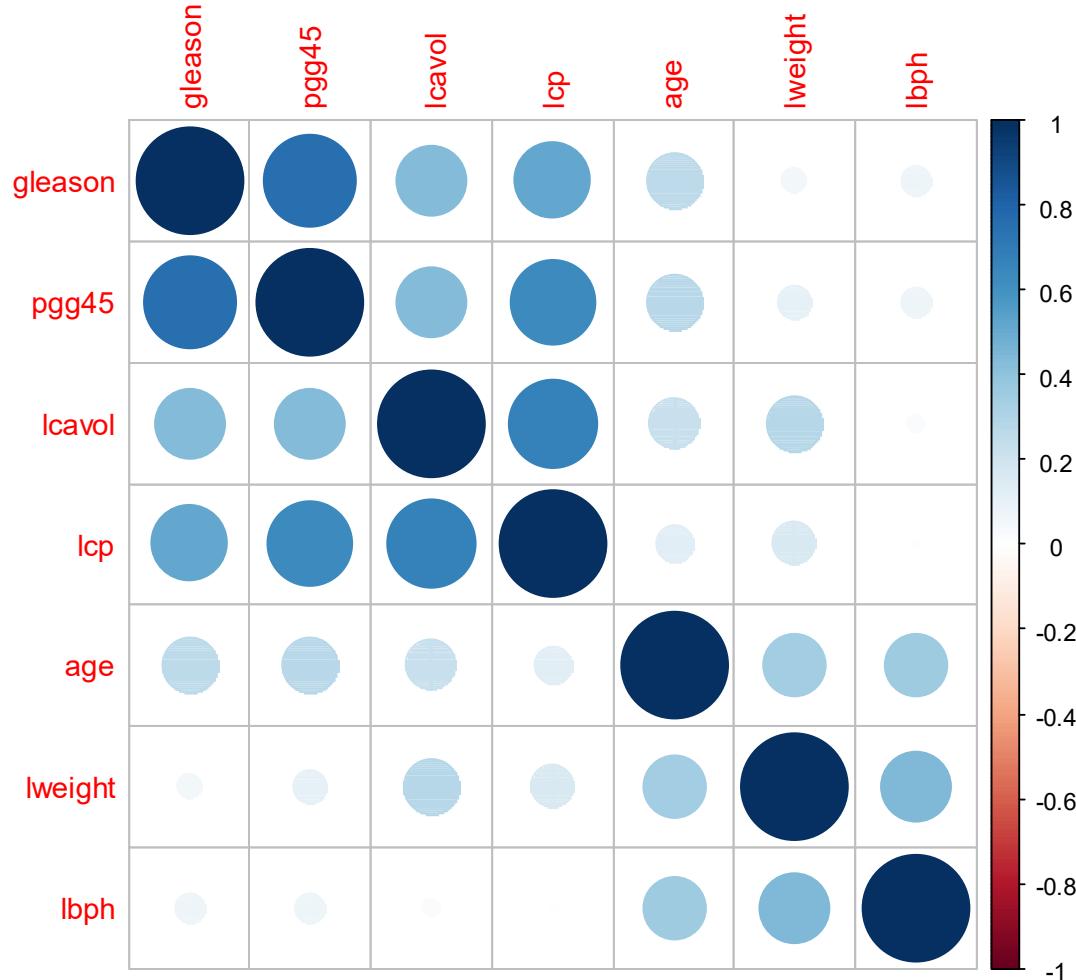
- Se va a realizar el ACP sobre las variables predictoras continuas
- Hay varias **correlaciones medias** ($r > 0.4$)

Ejemplo: Análisis de Componentes Principales



Ejemplo: Análisis de Componentes Principales

```
> ## Gráfico de la matriz de Correlaciones  
> library(corrplot)  
> dev.new()  
> corrplot ( cor.matr , order = "hclust" )
```



Ejemplo: Análisis de Componentes Principales

```
> ## Análisis de Componentes Principales (PCA) sobre la matriz R
> pca.out <- prcomp( xx[c(2:5,7:9)], scale=T )
> summary(pca.out)
Importance of components:
              PC1       PC2       PC3       PC4       PC5       PC6       PC7
Standard deviation   1.717   1.260   0.9291   0.79178   0.67116   0.57264   0.44218
Proportion of Variance 0.421 0.227 0.1233 0.08956 0.06435 0.04684 0.02793
Cumulative Proportion 0.421 0.648 0.7713 0.86087 0.92522 0.97207 1.00000
>
> ## Varianza explicada por cada CP. Valores propios (lambdas, eigenvalues)
> eigen <- pca.out$sdev ** 2
> eigen
[1] 2.9472572 1.5887704 0.8631677 0.6269137 0.4504552 0.3279132 0.1955227
> sum(eigen)
[1] 7
> 100 * ( eigen / sum(eigen) )    ## Comprobamos la varianza explicada por las CPs
[1] 42.103674 22.696720 12.330967  8.955910  6.435075  4.684474  2.793181
```

- Para realizar un **ACP** a partir de la matriz de **correlaciones R** usamos la función ***prcomp()*** con **scale=T** (Hay otra función que se llama ***princomp()***)
- La 1^a CP explica un 42.1% de la **variabilidad** total de los datos
- La 2^a CP explica un 22.7% de la variabilidad. Entre las 2 primeras CPs explican un 64.8% de la variabilidad. Las 7 CPs explican el 100% de la variabilidad
- Los **valores propios** se calculan como las varianzas de las componentes (SD^{**2})

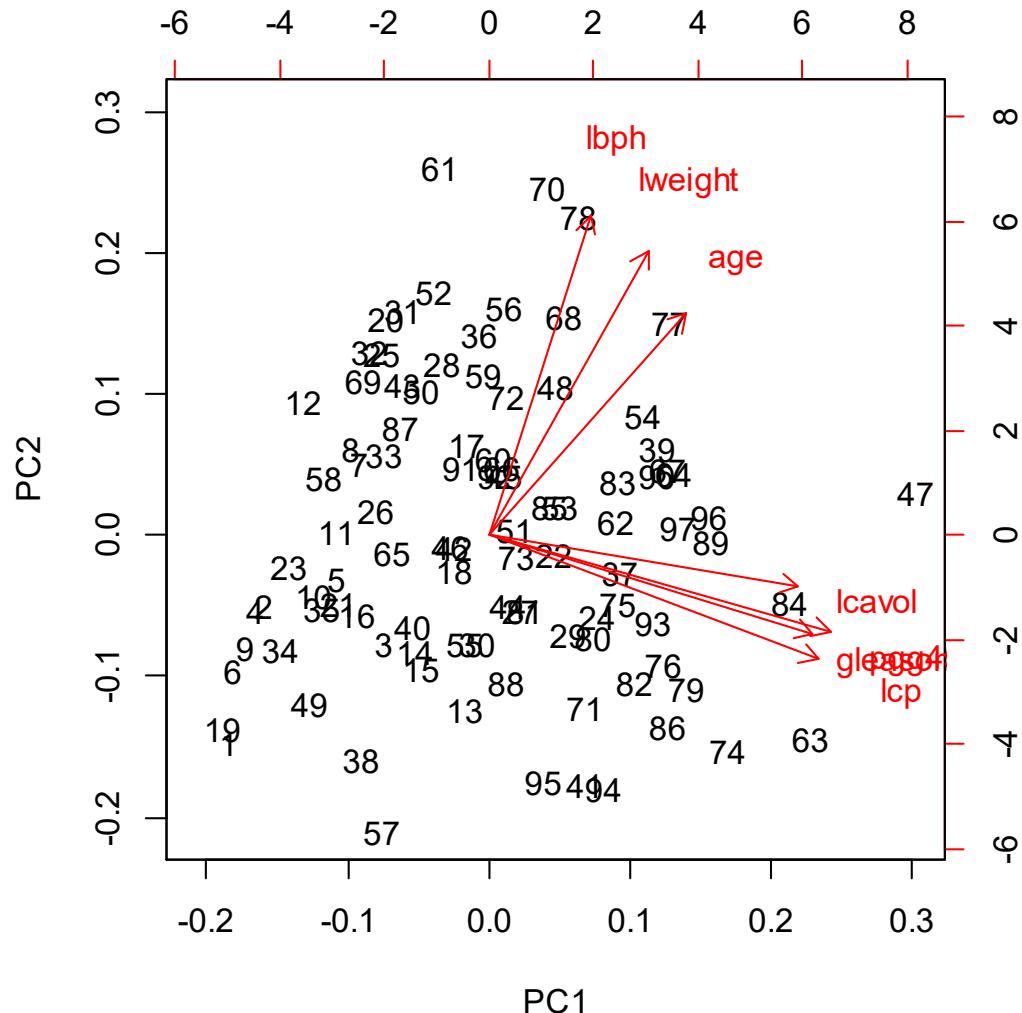
Ejemplo: Análisis de Componentes Principales

```
## Cargas Factoriales
> round ( pca.out$rotation , 3 )
    PC1   PC2   PC3   PC4   PC5   PC6   PC7
lcavol  0.437 -0.097  0.519 -0.215  0.365  0.488  0.337
lweight  0.225  0.549  0.440  0.072 -0.667  0.011 -0.058
age      0.279  0.427 -0.359 -0.750  0.132 -0.159 -0.078
lbph     0.143  0.615 -0.167  0.565  0.503  0.023  0.031
lcp      0.466 -0.239  0.315  0.097  0.211 -0.586 -0.479
gleason  0.458 -0.189 -0.418  0.157 -0.212  0.544 -0.463
pgg45    0.483 -0.187 -0.326  0.183 -0.248 -0.312  0.658
```

- Las **cargas factoriales**, extraídas del objeto **\$rotation**, representan las **correlaciones** entre las variables originales y las componentes, y pueden ser usadas para la **interpretación** de éstas
- La **1^a CP es explicada por las relaciones** que existen entre las variables lcavol, lcp, gleason y pgg45
- La **2^a CP es explicada por las relaciones** entre las variables lweight, age y lbph
- Los signos de las variables más importantes en cada CP son iguales (positivos en este caso); y eso indica que entre esas variables las **correlaciones son positivas**
- Las cargas pueden ser reportadas con estos mismos valores, pero distinto signo. No es importante ya que las CPs indican direcciones en el espacio de las Xs

Ejemplo: Análisis de Componentes Principales

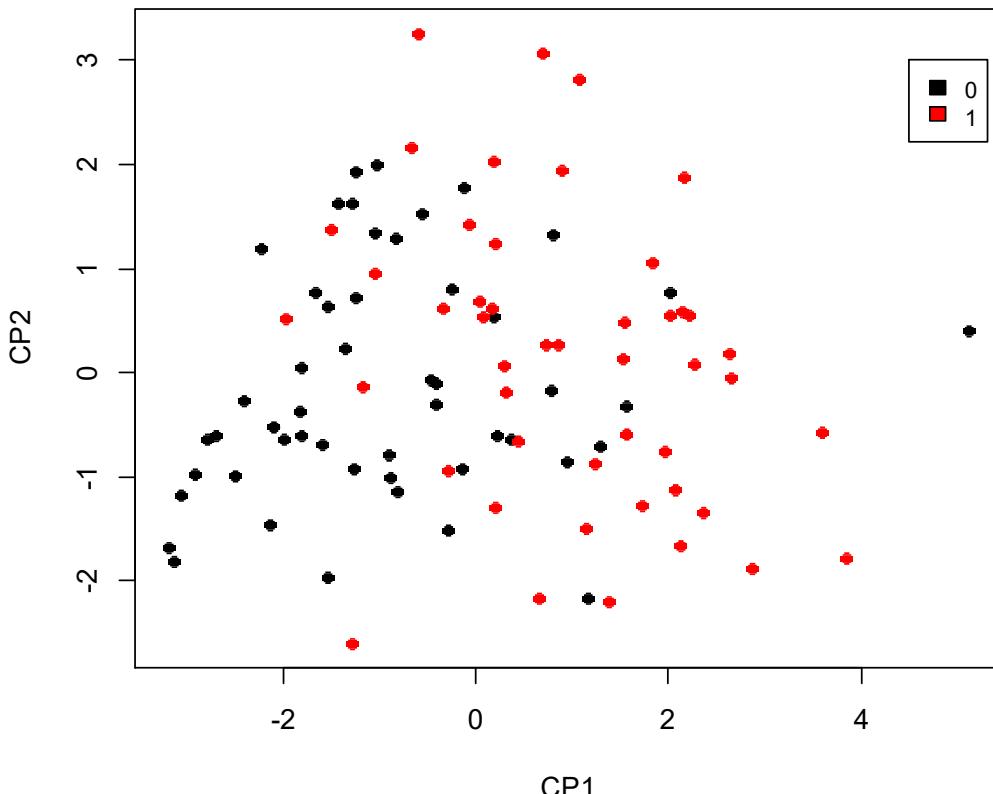
```
## Biplot  
dev.new()  
biplot(pca.out)
```



- La función **biplot()** muestra las observaciones y las variables en el **mismo gráfico** sobre las componentes principales
- Observaciones y variables están reescaladas de distinta forma
- Se pueden observar **los grupos de variables** que están asociadas a cada componente

Ejemplo: Análisis de Componentes Principales

```
> ## Gráfico de dispersión en las 2 primeras componentes
> dim(pca.out$x)
[1] 97 7
> cp1 <- pca.out$x [,1]
> cp2 <- pca.out$x [,2]
> cor(cp1, cp2)
[1] -1.541495e-16
> dev.new()
> plot( cp1, cp2, xlab="CP1", ylab="CP2", col=lpsa.gr, pch=16 )
> legend ( 4.5, 3 , leg=names(table(lpsa.gr)), palette() [1:2], cex=0.8)
```



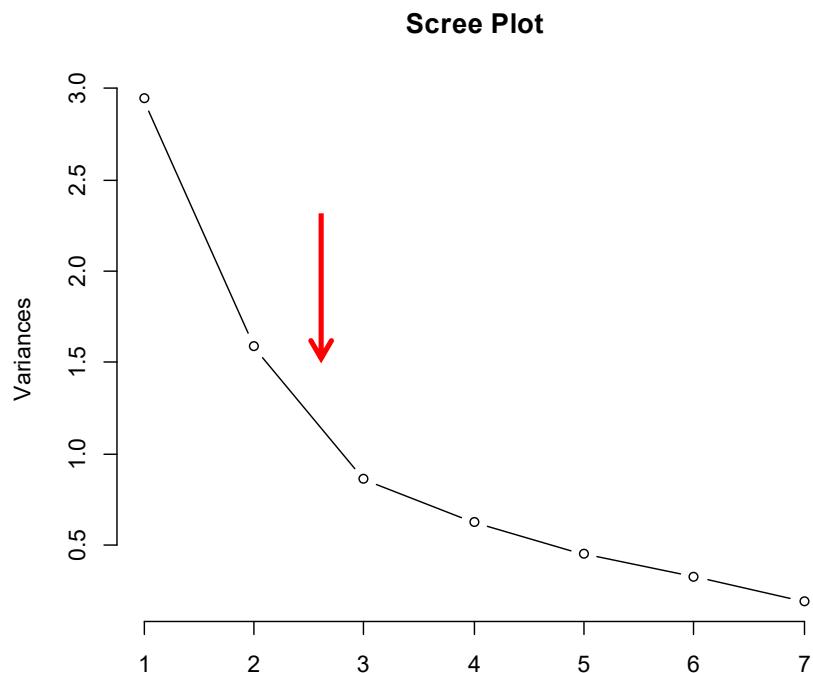
- El objeto `$x` contiene los valores de las observaciones en las CPs (scores, proyecciones)
- Se muestra un gráfico de dispersión en las 2 primeras CPs, con la variable `lpsa.gr` definiendo los colores
- La 1^a CP puede ser **importante** para distinguir entre los individuos que tienen Ilsa alto o bajo
- Sin embargo, la 2^o CP parece que no separa a estos grupos

Número de Componentes Principales

- Hay distintas formas de elegir el **número de componentes principales** que **explican adecuadamente la variabilidad** de los datos originales
- Fijar el **porcentaje de la varianza** que se desea explicar
 - Se suele elegir entre el **70% y el 90%**, dependiendo del problema concreto
- Seleccionar las componentes principales con **valores propios > 1**
 - Si las componentes se extraen a partir de R, se están usando las variables estandarizadas a varianza 1. Por tanto, con este criterio se están seleccionando CPs que **explican más varianza que la de una variable**
- **Gráfica “scree plot”** donde se representan los valores propios frente a las CPs ordenadas
 - Se busca un “codo”, un **punto de inflexión** en esta gráfica, y se seleccionan las CPs anteriores a ese punto de inflexión

Ejemplo: Número de Componentes Principales

```
> ## Número de Componentes Principales (Scree Plot)
> summary(pca.out)
Importance of components:
              PC1     PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation   1.717  1.260  0.9291  0.79178  0.67116  0.57264  0.44218
Proportion of Variance 0.421  0.227  0.1233  0.08956  0.06435  0.04684  0.02793
Cumulative Proportion  0.421  0.648  0.7713  0.86087  0.92522  0.97207  1.00000
> eigen
2.9472572 1.5887704 0.8631677 0.6269137 0.4504552 0.3279132 0.1955227
> dev.new()
> plot( pca.out, type ="l", main="Scree Plot")
```



- Las **2 primeras CPs** explican un **65% de la varianza**, lo cual se suele considerar un valor bajo. Con **3 CPs** se explicaría el **77%**
- Las **2 primeras CPs** tienen **valores propios** mayores que 1
- En el **scree plot** se observa un “codo” entre la **2^a** y **3^a** CPs, y por tanto, se deberían tomar **2 CPs**
- Se deberían elegir entre **2 o 3 CPs**

Ejemplo: PCA. Alta dimensionalidad

```
> ## Fichero Datos: ALLSubset
> xx2 <- read.delim("C://Data Mining con R/Datos/ALLSubset.txt", sep=" ")
> dim(xx2)
[1] 79 1001
> ## Análisis de Componentes Principales (PCA)
> pca.out.2 <- prcomp( xx2[1:1000], scale=T )
> summary(pca.out.2)

Importance of components:

          PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
Standard deviation   13.892 11.0045  7.58628  7.44111  6.03701  5.51194  5.19139  5.03819
Proportion of Variance 0.193  0.1211  0.05755  0.05537  0.03645  0.03038  0.02695  0.02538
Cumulative Proportion  0.193  0.3141  0.37163  0.42700  0.46344  0.49382  0.52077  0.54616

          . . . . .
          PC73      PC74      PC75      PC76      PC77      PC78      PC79
Standard deviation   1.5169 1.48067 1.45812 1.39228 1.37273 1.24780 2.657e-14
Proportion of Variance 0.0023 0.00219 0.00213 0.00194 0.00188 0.00156 0.000e+00
Cumulative Proportion  0.9903 0.99249 0.99462 0.99656 0.99844 1.00000 1.000e+00
> ## Loadings
> dim(pca.out.2$rotation)
[1] 1000 79
> ## Scores
> dim(pca.out.2$x)
[1] 79 79
```

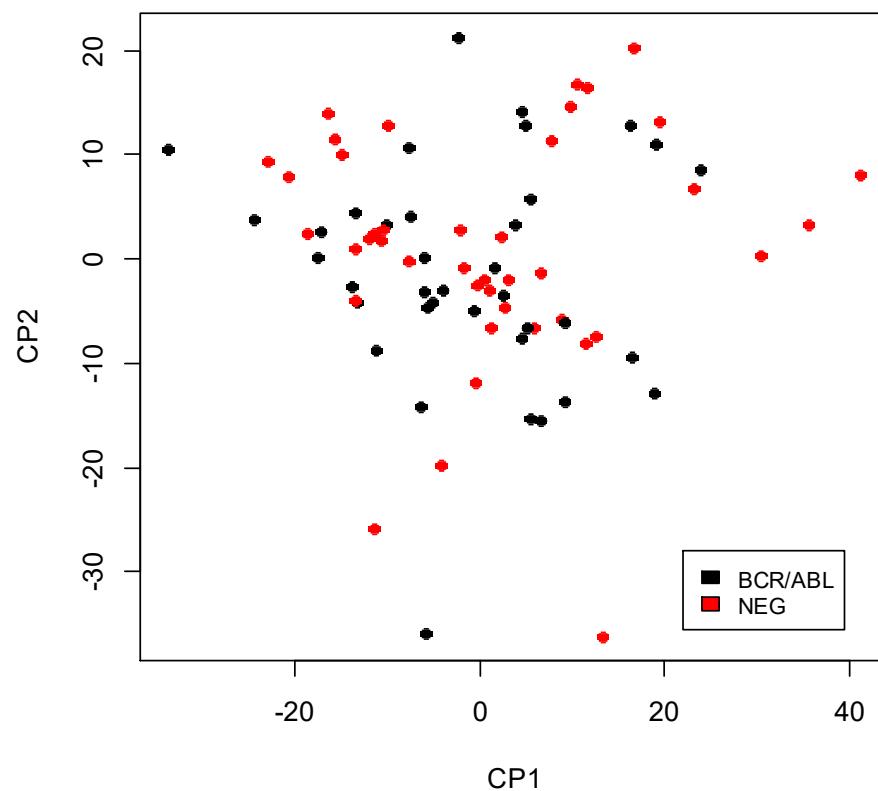
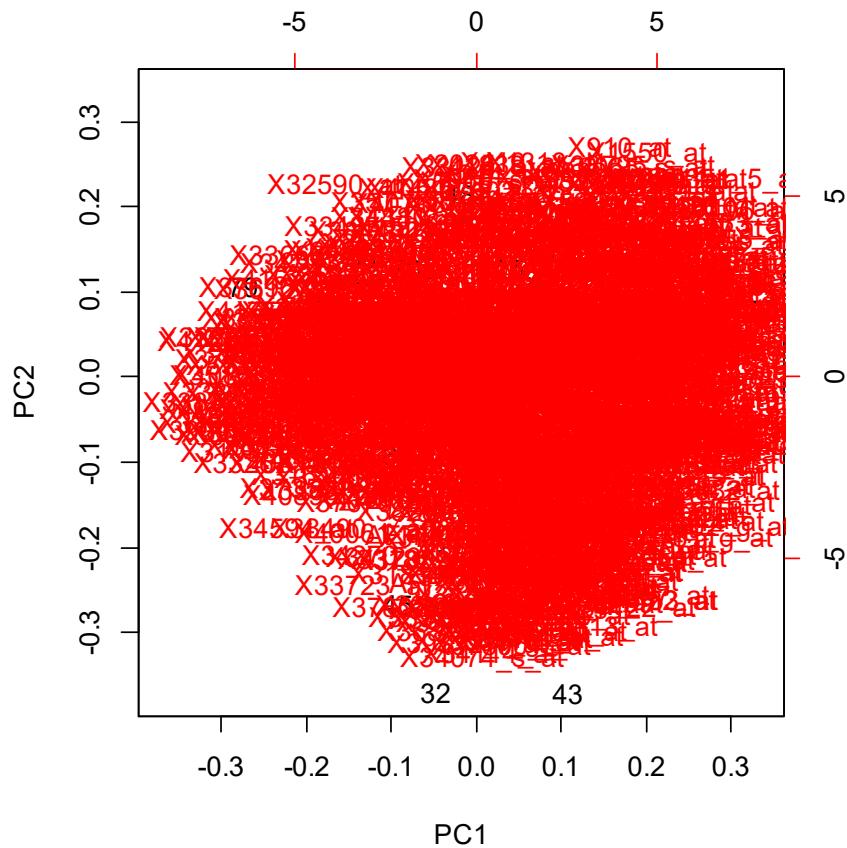
- Se han extraído **79 CPs (N)**. Las 2 primeras CPs explican el **31.4% de la varianza**
- Cada una de las 1000 variables tiene cargas en las 79 CPs; y cada observación tiene también un valor en cada una de las 79 CPs (proyecciones)

Ejemplo: PCA. Alta dimensionalidad

```

> ## Gráfico de dispersión en las 2 primeras componentes
> dev.new()
> biplot( pca.out.2 )
> dev.new()
> plot ( pca.out.2$x[ , 1], pca.out.2$x[ , 2] , xlab="CP1", ylab="CP2",
+         col=xx2$mol.biol, pch=16 )
> legend ( 22, -28, names(table(xx2$mol.biol)), palette(), cex=0.8 )

```



Componentes Principales (PCA). Observaciones

- Las covarianzas y las correlaciones entre variables solo tienen sentido para **variables continuas**, pero el Análisis de Componentes Principales es usado a veces con **variables ordinales y variables binarias**
 - Las variables dummy que definen una variable categórica son variables binarias
- Algunas técnicas predictivas **sustituyen las variables originales** por componentes principales, tratando de **eliminar información redundante**
 - Hay que tener en cuenta que para calcular cualquier componente principal son necesarias **todas las variables originales**
 - Por tanto, en este caso **no** se está haciendo una **selección de variables**

Ejercicio

- Fichero de datos: Virco
 - Variables a analizar: Todos los predictores [1:89]
- Realizar un **Análisis de Componentes Principales**
 - Mostrar un gráfico con la matriz de correlaciones (función *corrplot()*)
 - Calcular el número de CPs que son necesarias para explicar el 70% y el 99% de la varianza
 - Calcular el número de CPs que hay con varianza mayor que 1
 - Observación: Se está aplicando un ACP con variables binarias, lo cuál metodológicamente no es del todo correcto

Estadística Aplicada a la Investigación Biomédica con R

24 Análisis Cluster

- ✓ **Distancias entre observaciones**
- ✓ **Análisis Cluster Jerárquico**
- ✓ **Métodos No Jerárquicos. K-means**

Análisis Cluster

- El **Análisis Cluster** reúne una serie de técnicas estadísticas cuyo objetivo es la búsqueda de **grupos similares de observaciones**
 - Se usa para **clasificar** observaciones en grupos lo más homogéneos posible
 - También se conoce como **análisis de conglomerados**
 - De forma similar, el **análisis cluster de variables** se usa para agrupar variables
- El análisis cluster es una técnica multivariante **exploratoria y descriptiva**, es una **técnica no supervisada** ya que no hay ninguna variable que dirija el agrupamiento. Se está tratando de descubrir la **estructura de los datos**
- En general, para realizar un análisis cluster se necesita
 - Selección de **variables relevantes**
 - Selección de una medida de **proximidad entre observaciones**
 - Selección del **criterio para agrupar** observaciones en grupos

Distancias entre observaciones

- Todas la técnicas de análisis cluster están basadas en el cálculo de **distancias entre observaciones**. Las distancias más utilizadas son:
 - **Distancia euclídea**
 - Caso particular q=2
$$d(i, j) = \sqrt{\sum_{k=1, \dots, p} (x_{ik} - x_{jk})^2}$$
 - Distancia de Minkowski
 - Generalización
$$d_q(i, j) = \left(\sum_{k=1, \dots, p} |x_{ik} - x_{jk}|^q \right)^{1/q}$$
 - Distancia de Manhattan
 - Caso particular q=1
$$d_1(i, j) = \sum_{k=1, \dots, p} |x_{ik} - x_{jk}|$$
- Normalmente, las variables **se normalizan** antes de usar cualquier técnica de análisis cluster, para que en el cálculo de las distancias, todas las variables pesen lo mismo

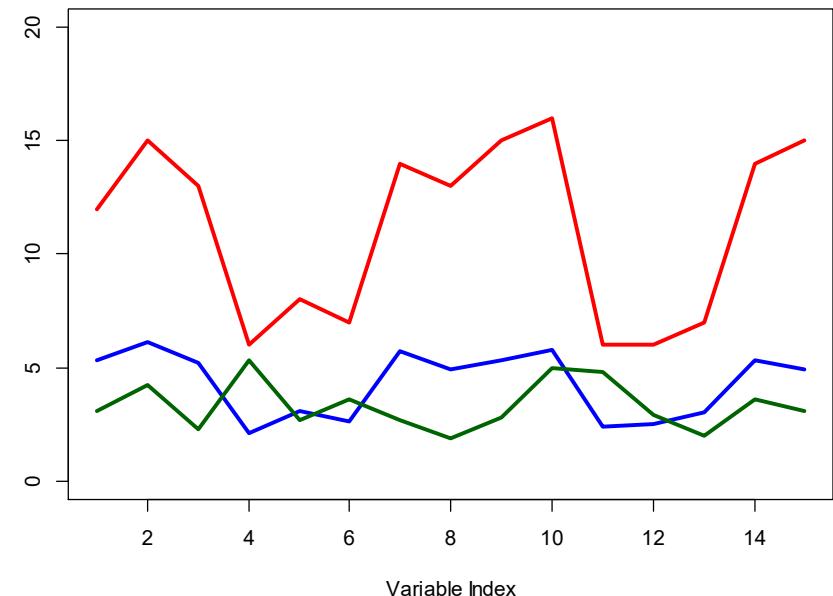
Distancias entre observaciones

- La **distancia basada en correlaciones** considera que dos observaciones son similares si sus variables están altamente correlacionadas, aunque sean observaciones que estén lejos en términos de distancia euclídea
- La **distancia** estará definida como $1 - R$

$$r(i, j) = \frac{\sum_{k=1}^p (x_{ik} - \bar{x}_i) \cdot (x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^p (x_{ik} - \bar{x}_i)^2 \cdot \sum_{k=1}^p (x_{jk} - \bar{x}_j)^2}}$$

La **correlación** es calculada entre 2 **observaciones** en lugar de 2 variables

- La distancia basada en correlaciones se enfoca en **la forma del perfil** de los valores de las observaciones en las variables
- Las observaciones de color rojo y azul están más correlacionadas, pero la verde y azul están más cerca en distancia euclídea



Análisis Cluster jerárquico aglomerativo

- El **Análisis Cluster jerárquico aglomerativo** tiene los siguientes pasos:
 - El análisis empieza con tantos grupos como observaciones
 - En cada paso, se calculan las distancias entre todos los grupos, y se unen los 2 grupos más próximos
 - Al final, todas las observaciones forman un único grupo
- En los análisis cluster jerárquicos, hay que decidir cuál es el **número óptimo de cluster** a formar, lo cual suele ser una decisión subjetiva
- Se visualizan con un árbol llamado **dendograma**, que muestra la estructura anidada del clustering, es decir, las uniones que se realizan en cada paso
- **Análisis Cluster jerárquico disociativo**
 - El proceso es el inverso, empezando con todos las observaciones en un cluster, y en cada paso se hace una partición hasta que cada observación forma un cluster

Distancias entre clusters

- En los distintos pasos de un análisis cluster jerárquico hay que definir la **distancia entre 2 grupos**, a la que se le llama “**linkage**”. Hay distintas opciones para definir la distancia entre los grupos A y B
 - **Distancia mínima (single linkage)**
 - Distancia entre los puntos más próximos
 - **Distancia máxima (complete linkage)**
 - Distancia entre los puntos más alejados
 - **Distancia media (average linkage)**
 - Media entre las distancias entre todos los puntos
 - **Distancia entre centroides (centroid linkage)**
- El método “**average linkage**” es uno de los más usados, y tiende a encontrar grupos de observaciones más balanceados

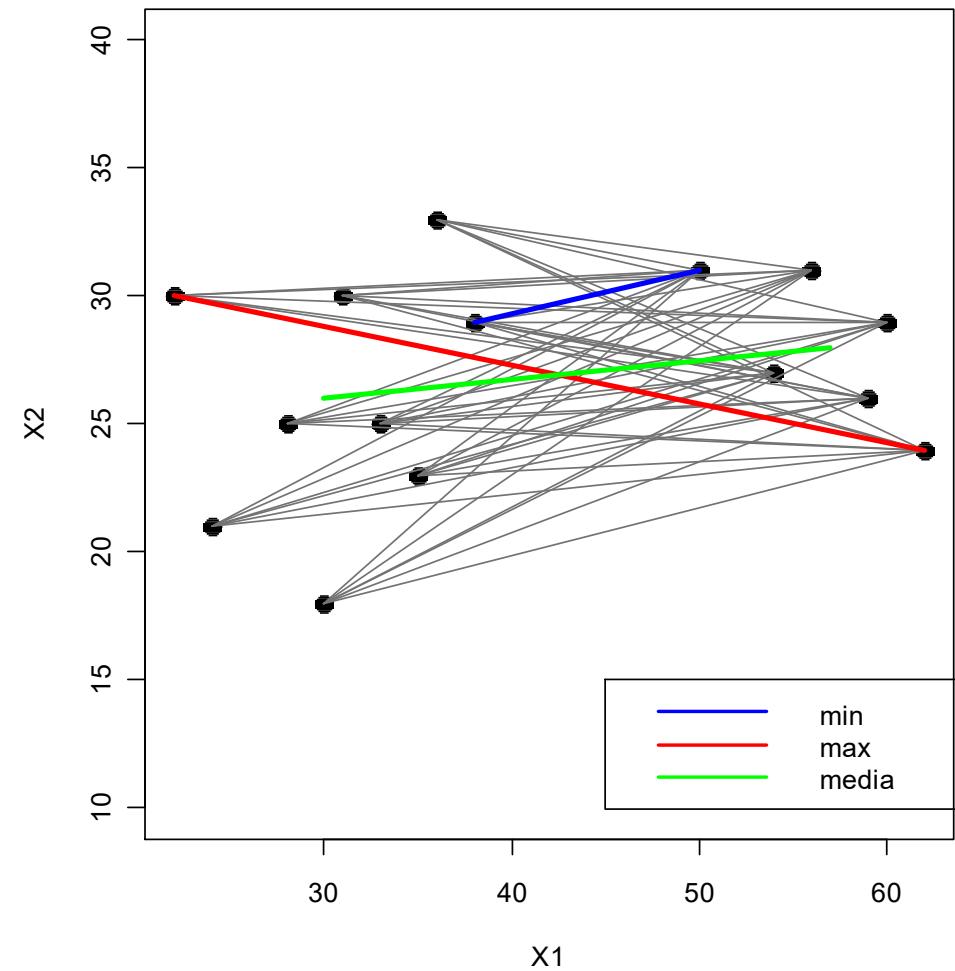
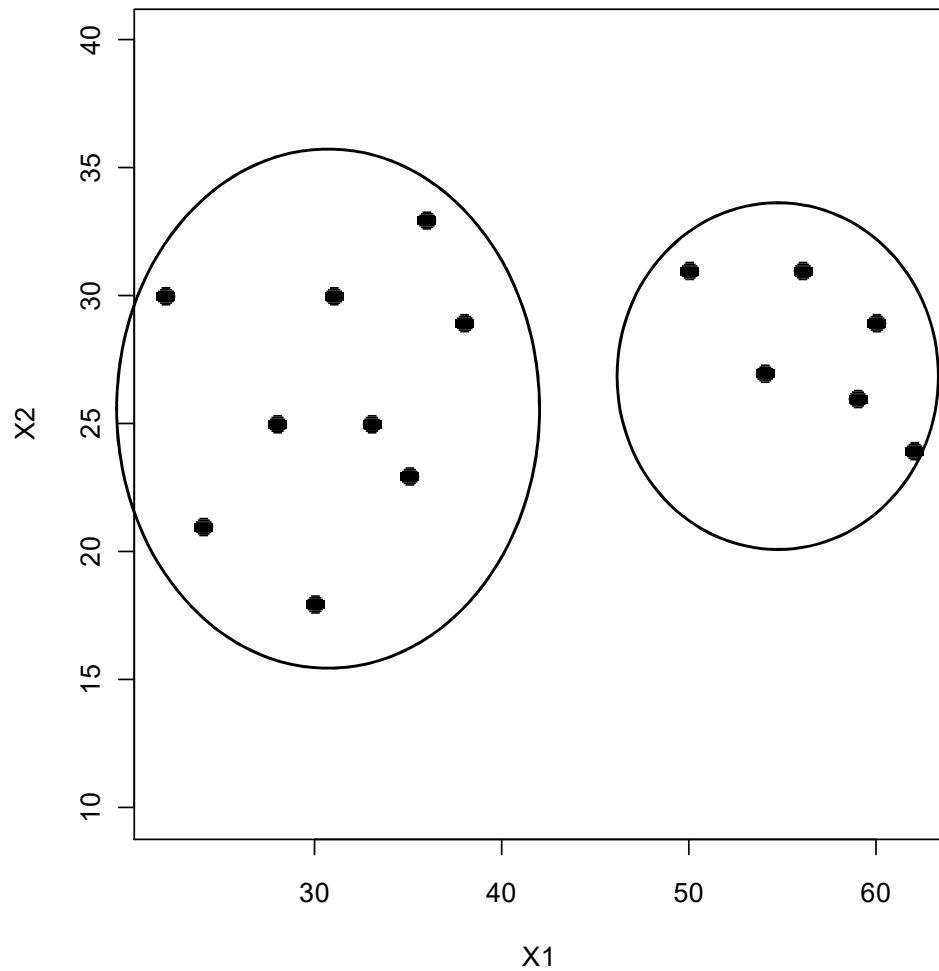
$$d_{AB} = \min_{i \in A, j \in B} (d_{ij})$$

$$d_{AB} = \max_{i \in A, j \in B} (d_{ij})$$

$$d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$$

$$d_{AB} = d(\bar{A}, \bar{B})$$

Distancias entre clusters



Ejemplo: Análisis Cluster jerárquico

```
> ## Fichero Datos: Microarray
> xx <- read.delim( "C://Data Mining con R/Datos/NCI Microarray.txt", sep=" ", header=F)
> dim(xx)
[1] 6830   64
> xx[1:3,1:10]
      V1      V2      V3      V4      V5      V6      V7      V8      V9      V10
1  0.300  0.679961  0.940  0.280  0.485  0.310 -0.830 -0.190  0.460  0.760
2  1.180  1.289961 -0.040 -0.310 -0.465 -0.030  0.000 -0.870  0.000  1.490
3  0.550  0.169961 -0.170  0.680  0.395 -0.100  0.130 -0.450  1.150  0.280
> yy <- read.delim( "C://Data Mining con R/Datos/NCI Microarray Type Tumors.txt", sep="
", header=F )
> dim(yy)
[1] 64   1
> head(yy)
      V1
1  CNS
2  CNS
3  CNS
4  RENAL
5 BREAST
> w.type.tum = yy$V1    ## Tipo de tumor (para etiquetar las observaciones)
```

- Cada **columna** de este fichero contiene datos de **expresión génica** de un tumor (una **observación**). Hay que **trasponer** los datos para analizarlos (función *t()*)
- Hay una variable que etiqueta el **tipo de tumor**, que está en otro fichero
- Los datos de expresión tienen la **misma escala**, y no se van a estandarizar

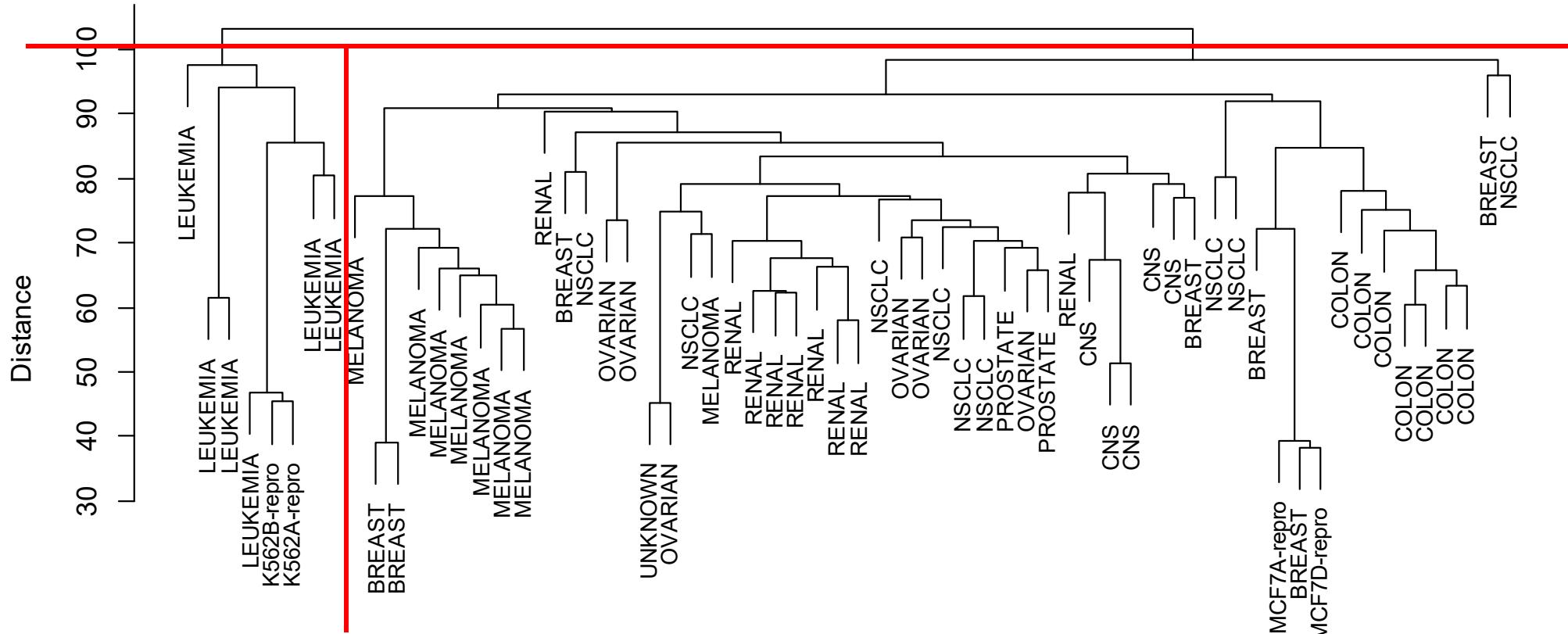
Ejemplo: Análisis Cluster jerárquico

```
> ## Análisis Cluster Jerárquico
>
> ## Single linkage (distancia mínima)
> clust.min <- hclust ( dist ( t(xx), method="euclidean" ) , method = "single" )
> ## Complete linkage (distancia máxima)
> clust.max <- hclust ( dist ( t(xx), method="euclidean" ) , method = "complete")
> ## Average linkage (distancia media)
> clust.med <- hclust ( dist ( t(xx), method="euclidean" ) , method = "average")
>
> ## Gráficos Dendogramas
> dev.new(width=16, height=8)
> par(mfrow = c(1,3) )
> plot( clust.min, main="Single linkage", lab=w.type.tum,
+       ylab="Distance", xlab="", sub="", cex=0.6 )
> plot( clust.max, main="Complete linkage", lab=w.type.tum,
+       ylab="Distance", xlab="", sub="", cex=0.6 )
> plot( clust.med, main="Average linkage", lab=w.type.tum,
+       ylab="Distance", xlab="", sub="", cex=0.6 )
```

- La función ***hclust()*** usa una **matriz de distancias** como parámetro, que se calcula con la función ***dist()*** y se aplica al dataframe de datos traspuesto para que las observaciones estén en las filas. Se usa la distancia euclídea
- La función ***plot()*** aplicada a un objeto *hclust* dibuja un **dendograma**
- Utilizamos el tipo de tumor para etiquetar las observaciones

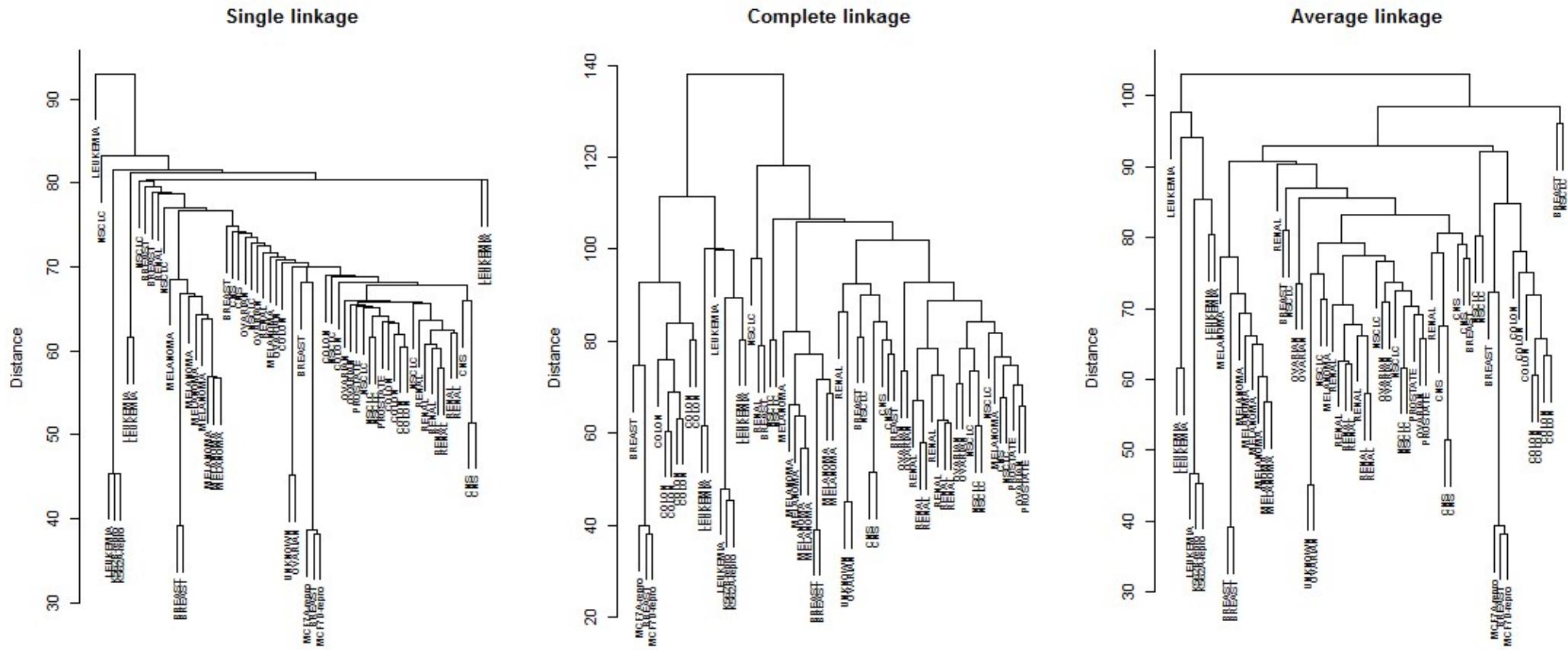
Ejemplo: Análisis Cluster jerárquico

Average linkage



- Desde abajo hacia arriba se va mostrando **el orden** en el que se hacen las uniones. Dos observaciones son próximas cuando su unión se produce en la parte baja del dendograma (en el eje vertical)
- Los **tumores del mismo tipo** aparecen juntos, lo que significa que la expresión de los genes de esos tumores tienen **características comunes**. Además, se puede ver qué tipos de tumores son más semejantes

Ejemplo: Análisis Cluster jerárquico



- Las agrupaciones son distintas según el método que se use. Cuando se usa “complete linkage” o “average linkage” los clusters tienden a ser más balanceados, mientras que “single linkage” tiende a proporcionar clusters más extendidos donde las observaciones se van uniendo una a una

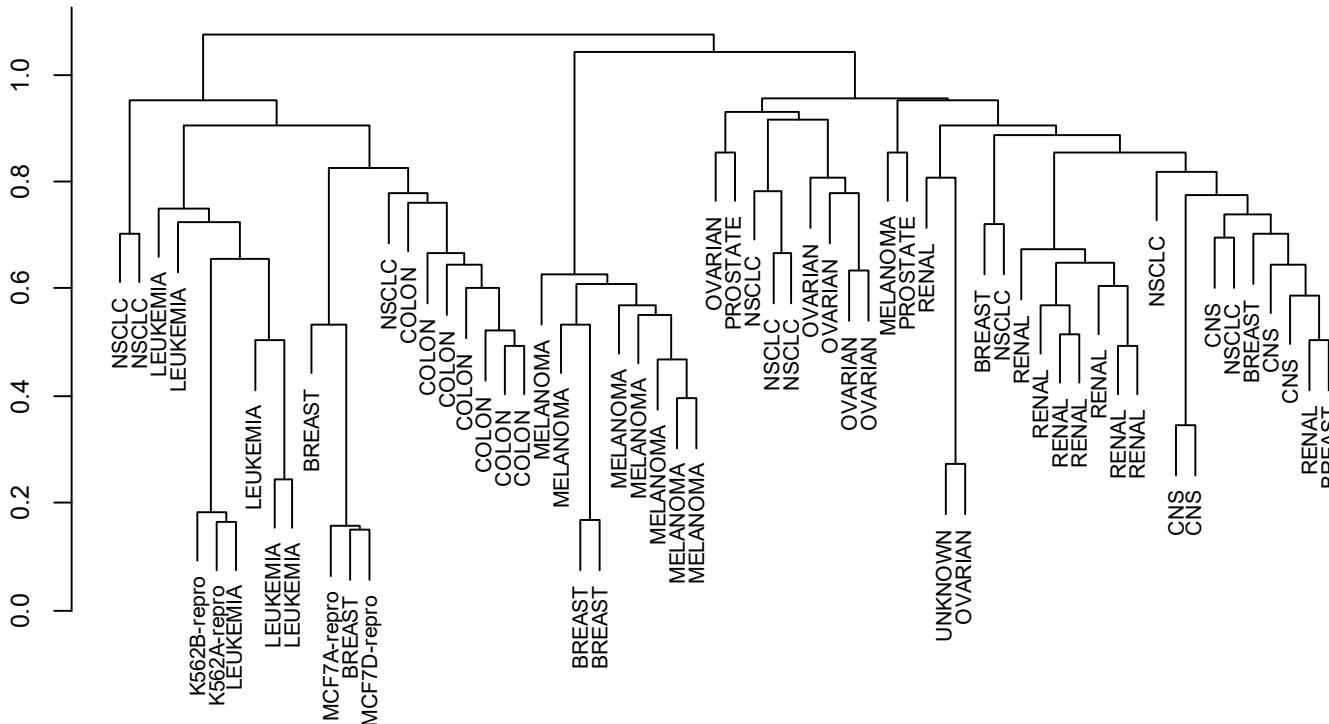
Ejemplo: Análisis Cluster jerárquico

```
> ## Obtener la solución con 2 clusters
> clust.2 <- cutree ( clust.med , k=2 )
> clust.2
   v1   v2   v3   v4   v5   v6   v7   v8   v9   v10  v11  v12  v13  v14  v15  v16  v17  v18  v19  v20  v21  v22
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
v23  v24  v25  v26  v27  v28  v29  v30  v31  v32  v33  v34  v35  v36  v37  v38  v39  v40  v41  v42  v43  v44
   1    1    1    1    1    1    1    1    1    1    2    2    2    2    2    2    2    2    2    2    1    1    1
v45  v46  v47  v48  v49  v50  v51  v52  v53  v54  v55  v56  v57  v58  v59  v60  v61  v62  v63  v64
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
> table(clust.2)
clust.2
  1   2
56   8
```

- La función ***cutree()*** se usa para generar una variable con el número de clusters que se deseen (parámetro ***k=***)
- Se definen 2 clusters, uno está compuesto por 56 observaciones y el otro con 8

Ejemplo: Análisis Cluster jerárquico

```
> ## Distancia basada en Correlaciones entre observaciones
> ## Importante: xx está ya traspuesto (observaciones en las columnas)
> w.corr.obs = cor ( xx , method = c("pearson") )
> dim(w.corr.obs)
[1] 64 64
> w.dist = as.dist( 1 - w.corr.obs )
> ## Average linkage (distancia media)
> clust.med.corr = hclust ( w.dist , method = "average")
> dev.new()
> plot(clust.med.corr, main="", lab=w.type.tum, ylab="Distance",xlab="",sub="",cex=0.7)
```

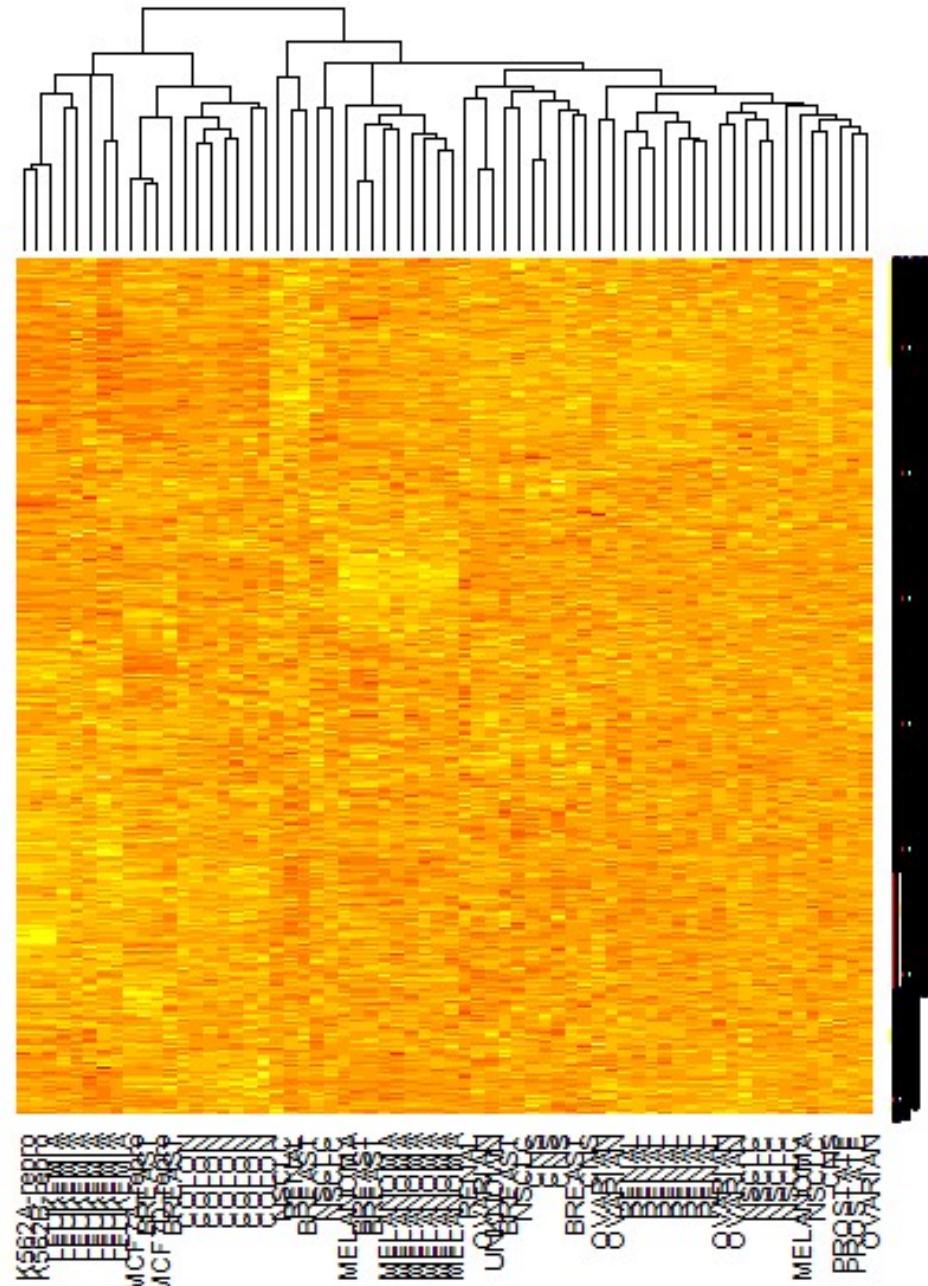


- Se calcula la matriz de correlaciones entre observaciones
- La **distancia** viene definida por **1 – R**, que se convierte con la función **as.dist()**
- El resultado obtenido es diferente a los anteriores

Ejemplo: Heatmap

```
> ## Heatmap  
> dev.new()  
> heatmap ( as.matrix(xx) ,  
           Rowv=NA , labCol=w.type.tum )
```

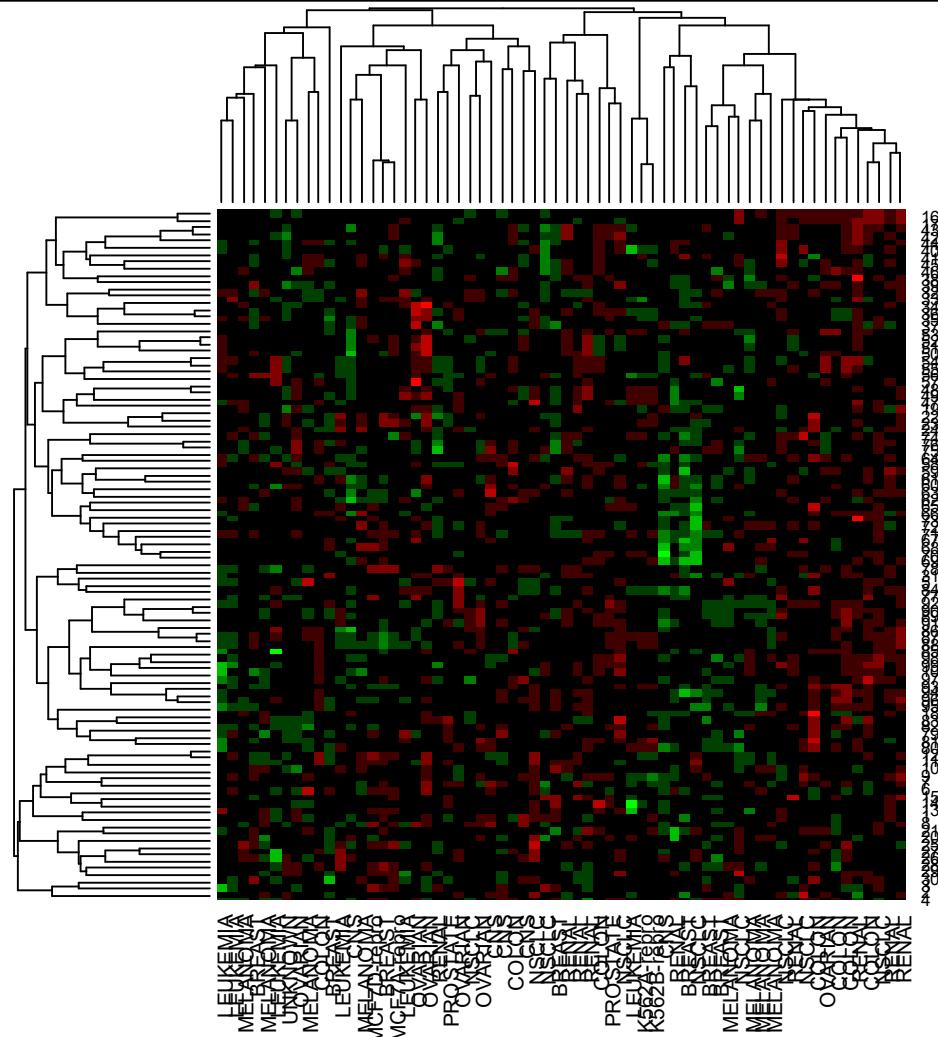
- La función **heatmap()** permite mostrar este gráfico, donde se agrupan observaciones después de un clustering
- Los datos hay que introducirlos como una matriz
- La opción *Rowv=NA* se usa para que no haga también el clustering por filas (variables)



Ejemplo: Heatmap

```
> library(gplots) ## para la escala de color  
> dist.pear <- function(x) as.dist ( 1 - cor(t(x)) )  
> hclust.ave <- function(x) hclust ( x , method="average" )  
> dev.new()  
> heatmap ( as.matrix( xx[1:100, ] ) , distfun=dist.pear, hclustfun=hclust.ave,  
+           labCol=w.type.tum, col=greenred(10) )
```

- Se muestra el gráfico con las 100 primeras variables
- Se selecciona la distancia y el *linkage* entre cluster
- Se usa la paleta de color “greenred” de la librería **gplots**



Métodos no jerárquicos

- Los métodos de **análisis cluster no jerárquicos** tienen como objetivo realizar una **única partición** de las observaciones en **k grupos**
 - El **número de clusters** que se van a formar debe ser especificado a priori
 - Se puede utilizar una técnica de análisis cluster jerárquico para determinar el número de clusters
 - Cada observación es asignada a un único cluster
- Se suelen llamar **métodos de optimización** porque se trata de buscar la partición de las observaciones que optimiza algún **criterio numérico**
- Suelen ser **métodos de reasignación** porque permiten que una observación asignada a un grupo en un determinado paso pueda ser reasignada a otro grupo en otro paso del algoritmo

Métodos no jerárquico. K-means

- El **método k-means** es uno de los más usados:
 - Se fija **k, el número de clusters**
 - Se elige una **partición inicial** en k grupos
 - Se pueden fijar k centroides de los grupos o elegirlos de forma aleatoria
 - Se **calculan las distancias** entre cada observación y cada centroide y **se asigna** cada observación al grupo al que está más cerca. Se suele usar la **distancia euclídea**
 - Se **recalculan los centroides** con todas las observaciones que hay en el cluster
 - El centroide de un cluster es la media de cada variable de las observaciones que lo forman
 - Se **repite el proceso** hasta que no haya reasignaciones, es decir, hasta que las observaciones no cambien de cluster

Métodos no jerárquico. K-means

- El método **k-means** pretende encontrar la partición de las observaciones en clusters que **minimicen la suma de las variaciones intra-grupo**

$$\min_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K W(C_k) \right\}$$

donde C_k representan los K clusters

- La **variación intra-grupo** $W(C_k)$ es una medida de la variación dentro de un cluster, de cómo todas las observaciones difieren entre sí. Se calcula como la suma de las distancias euclídeas al cuadrado de todos los pares de observaciones del cluster, dividido por el tamaño del cluster

$$W(C_k) = \frac{1}{n_k} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2$$

- Encontrar un óptimo global a este problema tiene mucho coste computacional
- El algoritmo **k-means** encuentra un **óptimo local**, que puede cambiar dependiendo de la **asignación inicial** de clusters

Ejemplo: Algoritmo k-means

```
> ## K-means con elección aleatoria de los centros
> km.out <- kmeans( t(xx) , cent=2 )
> km.out$cluster
   v1  v2  v3  v4  v5  v6  v7  v8  v9  v10  v11  v12  v13  v14  v15  v16  v17  v18  v19  v20  v21  v22
   1   1   1   1   1   1   1   1   1   1    1   1    1   1    1   1    1   1    1   1    1   1    1   1
v23 v24 v25 v26 v27 v28 v29 v30 v31 v32 v33 v34 v35 v36 v37 v38 v39 v40 v41 v42 v43 v44
   1   1   1   1   1   1   1   1   1   1    2   2    2   2    2   2    2   2    2   2    2   2    2   2
v45 v46 v47 v48 v49 v50 v51 v52 v53 v54 v55 v56 v57 v58 v59 v60 v61 v62 v63 v64
   2   2   2   2   2   2   2   1   2   2    1   1    1   1    1   1    1   1    1   1    1   1    1
> table(km.out$cluster)
  1   2
43  21
> table(km.out$cluster, w.type.tum)
  w.type.tum
  BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro MCF7D-repro MELANOMA
  1      5   5     0          0          0          0          0          0      8
  2      2   0     7          1          1          6          1          1      0
  w.type.tum
  NSCLC OVARIAN PROSTATE RENAL UNKNOWN
  1      7     6     2     9     1
  2      2     0     0     0     0
```

- La función ***kmeans()*** ejecuta el algoritmo **k-means**. Si el parámetro **cent=** se iguala a un número, indica el número de clusters que se van a formar y asigna los centroides aleatoriamente
- Si el algoritmo k-means se ejecuta varias veces, puede dar resultados distintos
- La mayoría de los tumores del mismo tipo han caído en el mismo cluster

Ejemplo: Algoritmo k-means

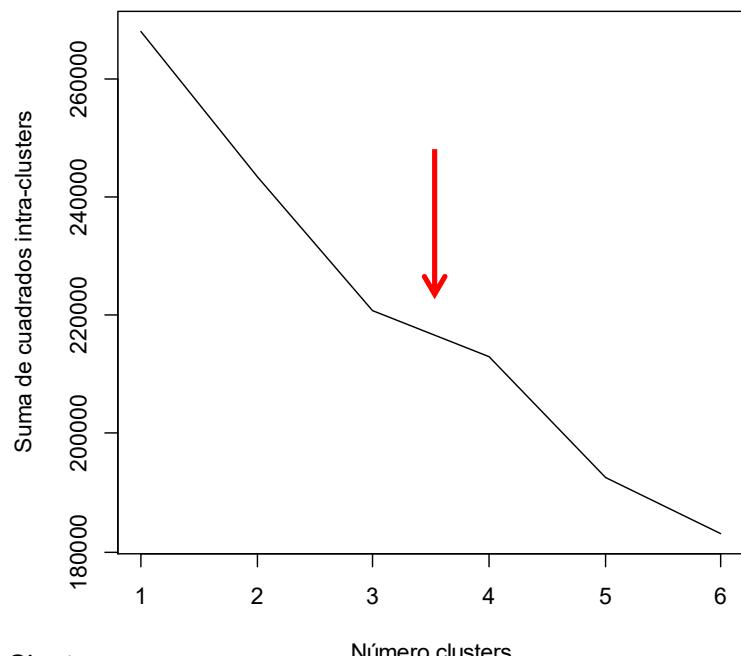
```
> ## K-means eligiendo los centros
> centros <- matrix ( NA, 2, nrow(xx) ) ## 2 centros, 6830 variables
> length ( xx [ , 1 ] )
[1] 6830
> centros[1, ] <- xx [ , 10 ]
> centros[2, ] <- xx [ , 20 ]
> centros [ , 1:10]
     [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]      [,10]
[1,]  0.76 1.49 0.28 0.10 -0.525 0.36  0.6  0.175 0.58  1.1450190
[2,] -0.29 0.00 0.05 0.73  0.385 0.39  0.0 -0.065 -0.73 -0.8849805
>
> ## K-means
> km.out.2 <- kmeans( t(xx) , cent = centros )
>
> ## Comprobando los resultados con el anterior cluster
> table ( km.out$cluster, km.out.2$cluster )

    1   2
1 34  9
2  0 21
```

- Con el parámetro **cent=** se pueden definir los centros. Una opción fácil es elegir 2 observaciones como centros (columnas 10 y 20) que sepamos que no están cerca
- Hay un grupo de 9 observaciones que antes estaban en el cluster 1 y ahora están en el cluster 2

Ejemplo: Número de clusters óptimo

```
> ## Número óptimo de clusters. Gráfico probando entre k=2 y k=6
> withinss.sum <- rep(NA, 6)
>
> ## SS de todos los datos, antes del clustering: es la suma de las varianzas de
> ## todas las variables multiplicado por el número de observaciones - 1
> n <- ncol(xx)
> withinss.sum[1] <- sum ( apply ( xx, 1 , var ) ) * ( n - 1 )
>
> for ( k in 2:6 )
+ { withinss.sum[k] <- kmeans( t(xx) , cent = k )$tot.withinss      }
> # Gráfico de SS
> dev.new()
> plot( 1:6, withinss.sum, type="l", xlab="Número clusters",
+           ylab="Suma de cuadrados intra-clusters")
```



- Entre las soluciones de 3 y 4 clusters parece que hay un punto de inflexión, un “codo”
- Según, este gráfico el **número óptimo de clusters** es **3**
- Este gráfico puede cambiar porque k-means está ejecutado con centros aleatorios

Ejemplo: Algoritmo k-means

```
> ## K-means con K=3 ejecutado 20 veces
> km.out <- kmeans( t(xx) , cent=3 , nstart=20 )
> table(km.out$cluster)

 1  2  3
34 21  9

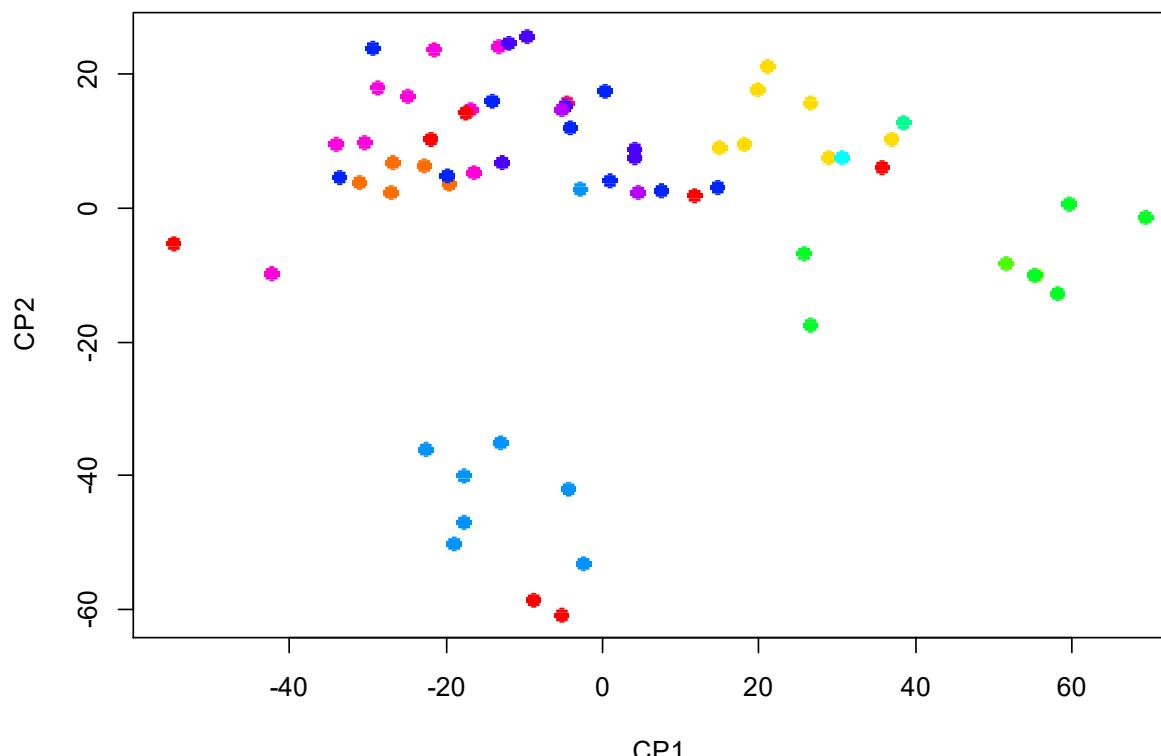
> table(km.out$cluster, w.type.tum )
  w.type.tum
    BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro MCF7D-repro MELANOMA
 1      3   5     0         0         0         0         0         0       1
 2      2   0     7         1         1         6         1         1       0
 3      2   0     0         0         0         0         0         0       7

  w.type.tum
    NSCLC OVARIAN PROSTATE RENAL UNKNOWN
 1      7     6     2     9     1
 2      2     0     0     0     0
 3      0     0     0     0     0
```

- Se ejecuta la función ***kmeans()*** con el parámetro ***nstart*=** que indica el número de veces que se va a ejecutar el algoritmo
- Solo se reporta el **mejor modelo** de las 20 ejecuciones, es decir, se muestra el modelo con el **mejor óptimo local** entre las 20 ejecuciones, con menor variación intra-grupo
- La mayoría de los tumores del mismo tipo han caído en el mismo cluster

Ejemplo: Algoritmo k-means. Gráfico con CPs

```
> ## PCA
> pca.out <- prcomp( t (xx) , scale=T )
> ## Scores
> dim(pca.out$x)
[1] 64 64
> Cols = function (vec)
+ { cols = rainbow(length(unique(vec)))
+   cols [ as.numeric ( as.factor (vec)) ]
+ }
> dev.new()
> plot ( pca.out$x[ , 1], pca.out$x[ , 2] , xlab="CP1", ylab="CP2", cex=1.2,
+       col=Cols(w.type.tum) , pch=16 )
```



Ejercicio

- Fichero de datos: ALLSuset (expresión de genes)
 - Variables a analizar: Todos los predictores [1:1000]
- Realizar un **Análisis Cluster Jerárquico** con la distancia media (average linkage)
 - Mostrar un dendograma, etiquetando las observaciones con la variable respuesta (*lab=xx\$mol.biol*)
- Realizar un **Análisis Cluster** con el algoritmo **k-means** y 2 clusters
- Realizar un Análisis de Componentes Principales; y mostrar las observaciones en un gráfico en la 1^a y 2^a CPs; y en otro gráfico en la 5^a y 7^a CPs , etiquetando en ambos las observaciones con la variable respuesta (*lab=xx\$mol.biol*)